MUNDO ATARI®

Número 1

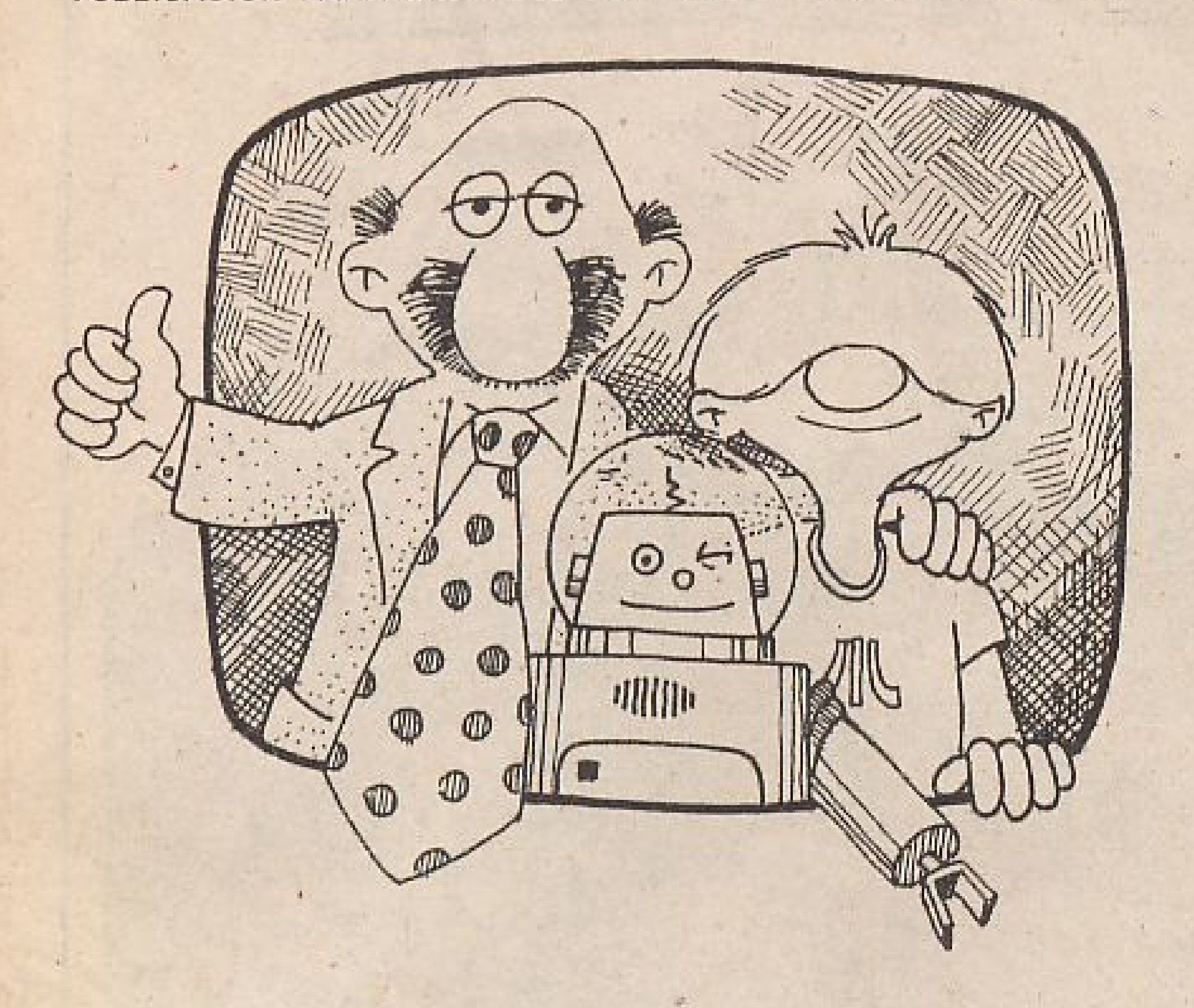
JUNIO 1987

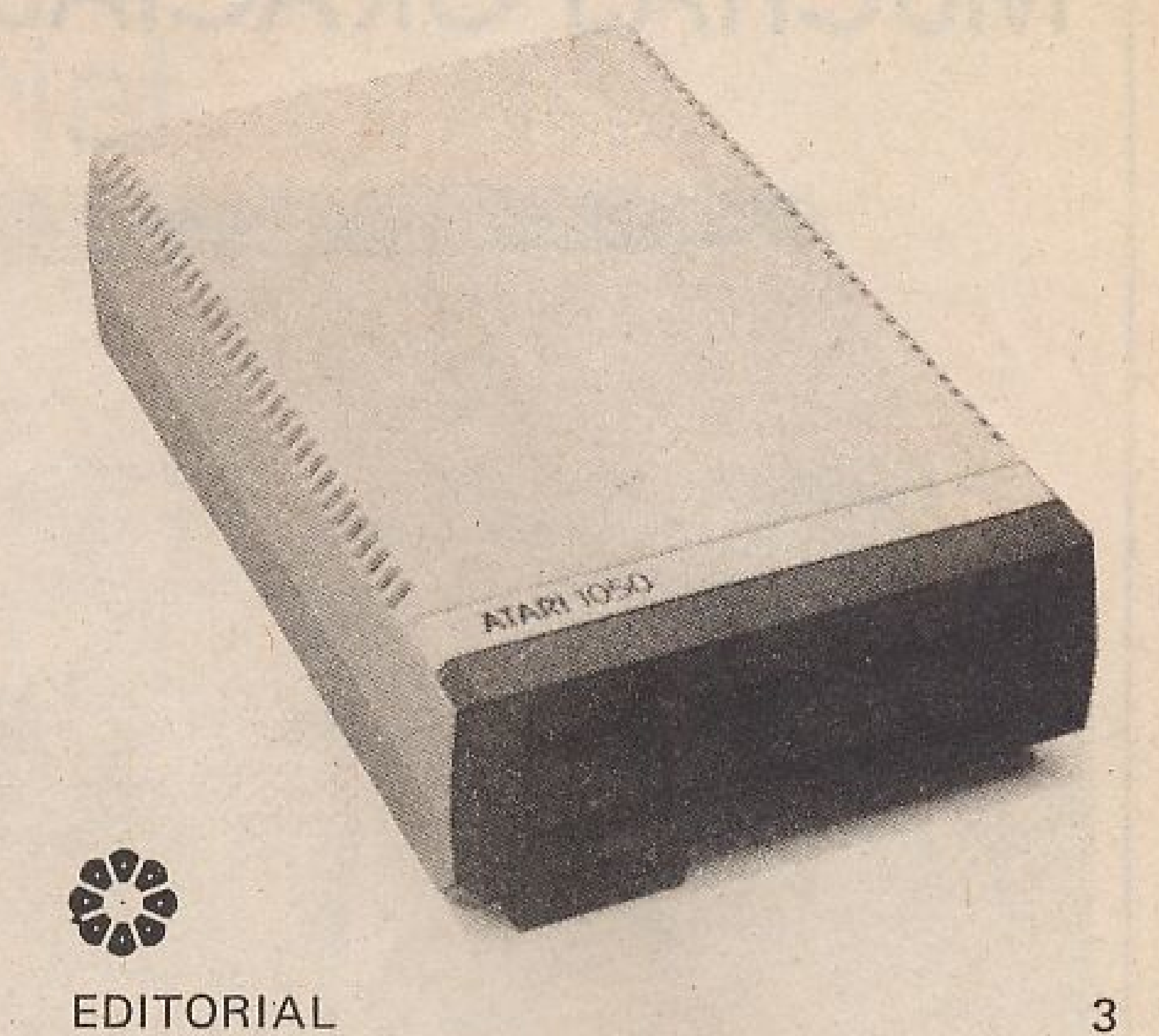
$ 300

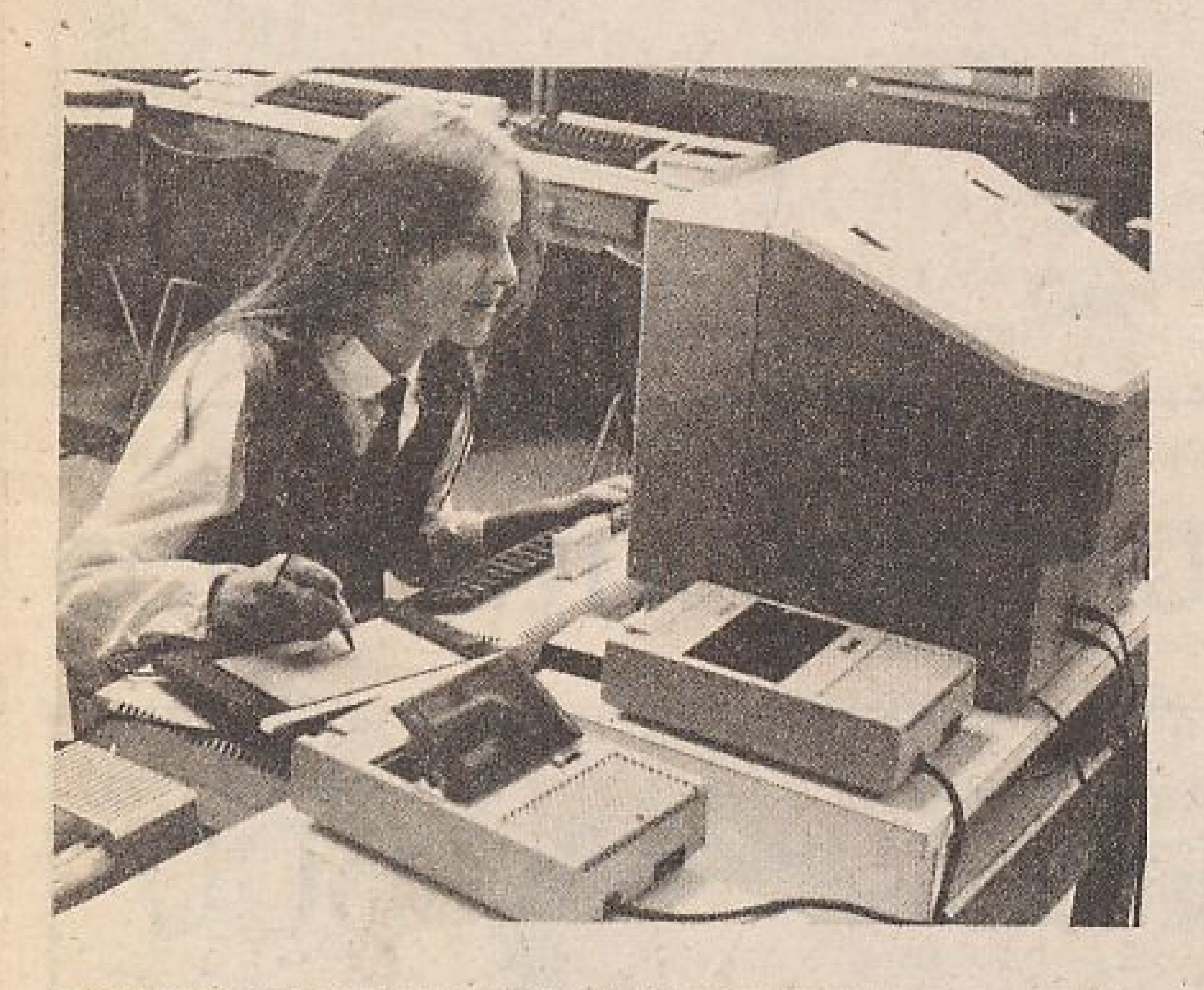
PUBLICACION PARA USUARIOS DE MICROCOMPUTADORES ATARI

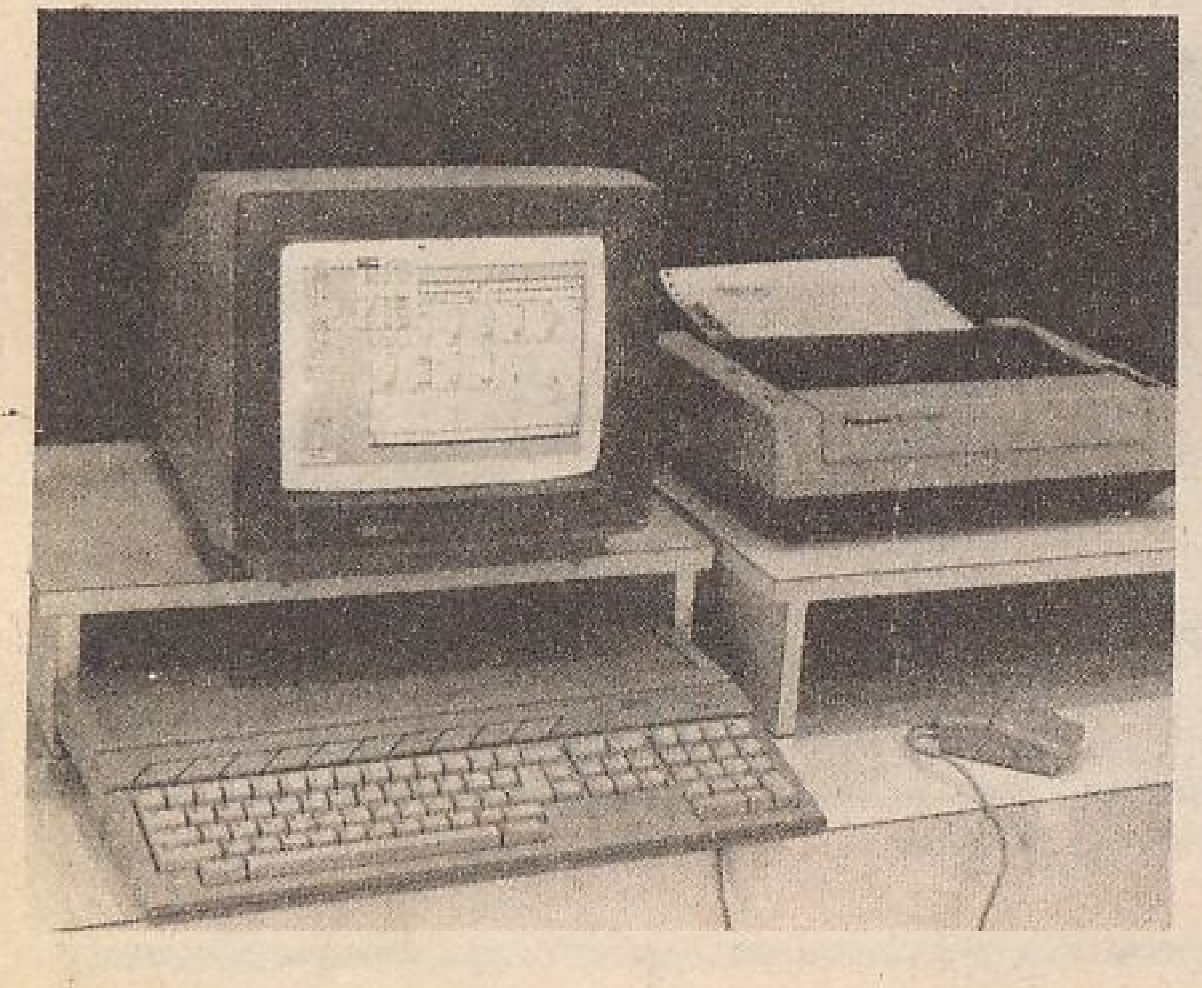
TU REVISTA AMIGA

MUNDO ATARI®
PUBLICACION PARA USUARIOS DE MICROCOMPUTADORES ATARI





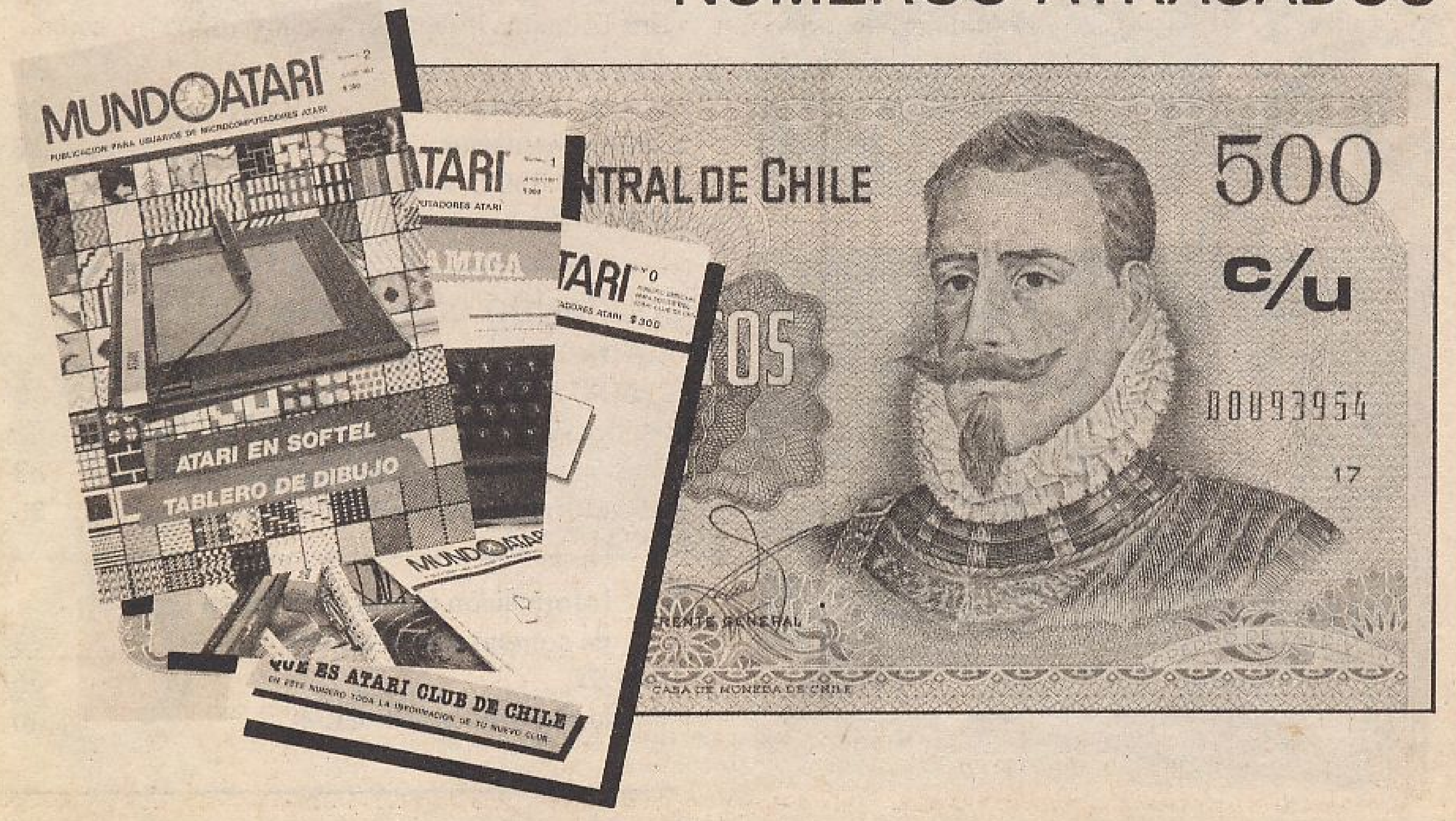




MUCHAS GRACIAS

A todos nuestros lectores que nos han estado estimulando con sus expresiones de apoyo por la reciente aparición de MUNDOATARI.

NUMEROS ATRASADOS



Editorial

TODO lo que podamos decir acerca de objetivos e intenciones de nuestra revista MUNDOATARI será capaz de modificar los juicios y opiniones que usted se forme con la sola lectura de las páginas siguientes.

En estas páginas reflejamos nuestra voluntad de proporcionar a los usuarios latinoamericanos de los microcomputadores ATARI un producto ameno y de alta calidad.

Recordemos que hasta hace pocos años la Informática era una ciencia de enormes equipos manipulados por un grupo selecto de personas. Sus términos eran incomprensibles para el resto de la gente.

Hoy sin embargo, se encuentra al alcance incluso de los niños de corta edad, que interactúan en base a programas diseñados para diferentes objetivos, confirmando un texto famoso en relación a los microcomputadores: *es la única herramienta ante la cual es imposible no aprender.*

Lo realmente importante es que el microcomputador ATARI es una magnífica herramienta de experimentación y diversión a su alcance.

El formato, definición y estilo de MUNDOATARI están orientados a cumplir con esta premisa tan elemental. Para ello entregamos en las diferentes columnas conceptos, actividades y normas para posibilitar el perfeccionamiento de la persona en los diferentes niveles.

Una palabra para los hogares, lugar inicial de aprendizaje. Nuestro ATARI puede y debe llegar a ser un motivo de reunión del núcleo familiar, de modo que la familia pueda integrarse y lograr una proyección de unidad en el tiempo .

Amigo lector, la ayuda que usted pueda proporcionarnos: sugerencias, artículos, programas, publicidad, suscripciones, etc. la recibiremos como estímulo para continuar con esta tarea que acabamos de iniciar.

Hasta el próximo mes.



JUNIO 1987 **Precio: $ 300**

Revista con información exclusiva para microcomputadores ATARI

Resol. Exenta No. 360/6-5-1987

Editor: Iván Gjurovic M.
Director: Adolfo Torrejón S.
Representante legal: Lucía Segura G.
Producción: SES Sistema
Casilla: 458, Correo Ñuñoa, Santiago
Teléfono: 2256579

Impreso por Editorial Antártica quien sólo actúa como impresora

Esta revista no mantiene relación de dependencia de ningún tipo con respecto a los fabricantes de microcomputadores ATARI ni sus representantes.

Solicite demostración de este programa en Centros ATARI y distribuidores autorizados a lo largo del país

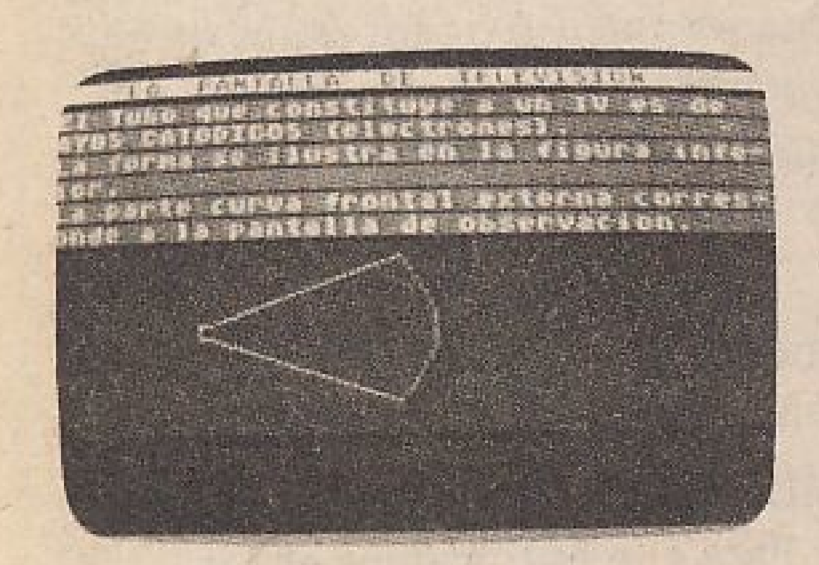
Técnicas

Las características gráficas del ATARI son particulares y han permitido un hecho histórico si consideramos la velocidad de cambios en el hardware: **la permanencia sostenida en el tiempo.**

Su estructura lo hace indicado para la educación. Prueba de ello es la cantidad de colegios chilenos que trabajan con esta marca en sus aulas (ver listado de colegios en MUNDOATARI No. 0).

Las posibilidades de crear y mejorar programas para los atarianos chilenos aumenta con la ayuda de una línea didáctica de software: TECNICAS.

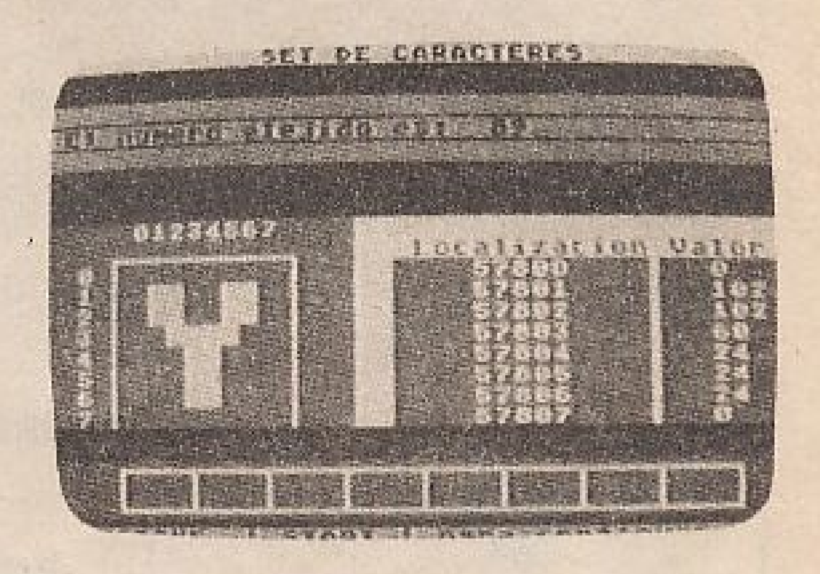
Este simple nombre identifica a seis programas en diskette, dirigidos a aquellos autodidactas que quieran profundizar en los conocimientos sobre su computador ATARI.



1. LISTA DE DESPLIEGUE:
Proporciona información sobre los procesos internos del receptor de TV, instrucción GRAPHICS y Modos Gráficos, ubicación de la Lista de Despliegue en memoria, instrucciones que puede contener una Lista y modificaciones a la misma.



2. SCROLLING:
Este programa informa acerca de la relación entre pantalla física y la ubicación en memoria, Scrolling y mapa de memoria, Scrolling y Lista de Despliegue. Informa además sobre Scrolling horizontal y Scrolling vertical.



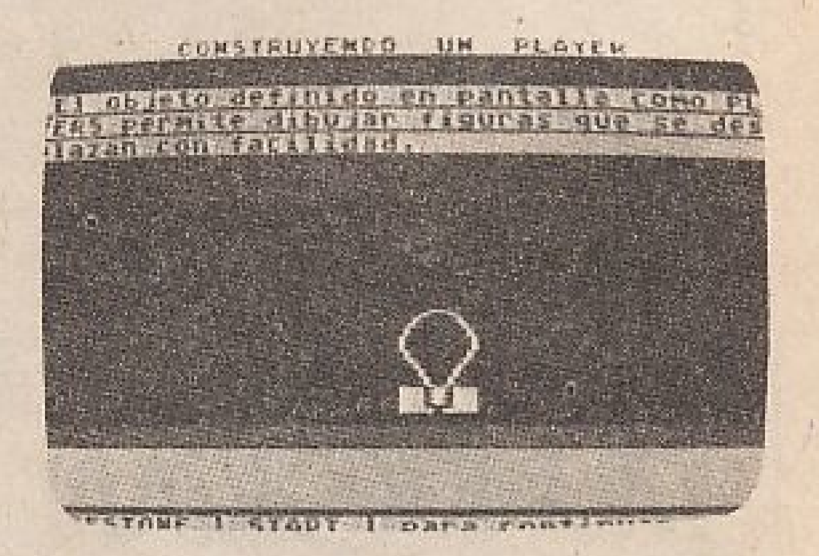
3. SET DE CARACTERES:
Permite utilizar la técnica de la modificación del set de caracteres para el desarrollo de programas que implican animación.



4. INTERRUPCION DEL VERTICAL BLANK:
Esta técnica permite optimizar el movimiento de Player, Scrolling, Flipping. Explica en particular una rutina para ubicar en la interrupción.



5. INTERRUPCION DE LISTA DE DESPLIEGUE:
Técnica que permite disponer de una mayor cantidad de colores en pantalla que los usuales.



6. PLAYER MISSILE:
Programa que da a conocer las bases sobre la animación de figuras denominadas Player.

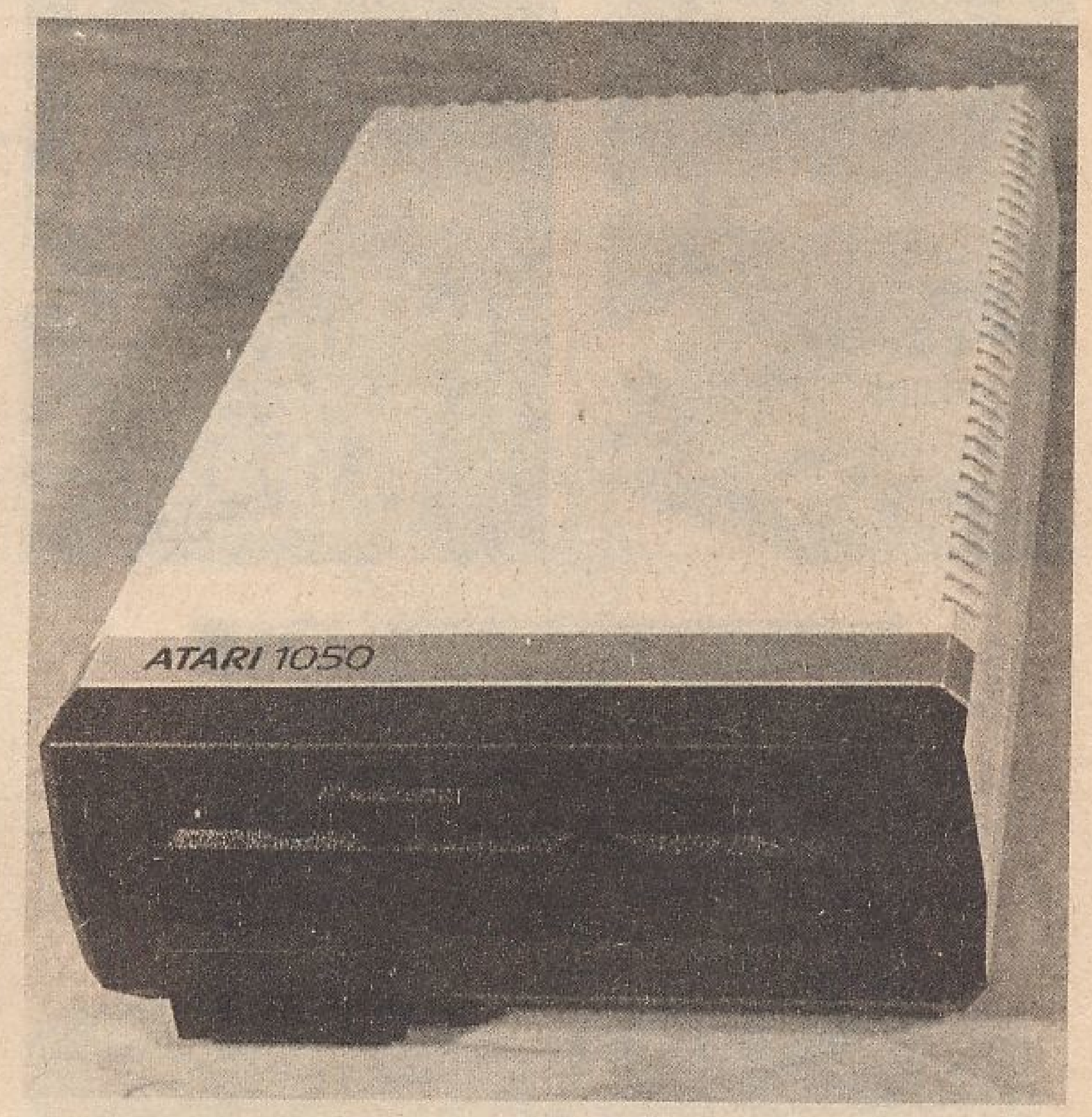
Unidad de Disco ATARI 1050

La unidad de diskette ATARI 1050 es el paso lógico que deben seguir todos los usuarios de equipos computacionales ATARI 800 XL ó 130 XE, que quieran ampliar las capacidades dè su equipo y obtener nuevas e interesantes aplicaciones.

Este periférico es un eficiente mecanismo de almacenamiento de información en un medio magnético, como son los diskettes de 5 1/4 pulgadas que maneja, permitiendo agilizar los procesos de lectura de programas de gran tamaño, como también agilizar la escritura de archivos de datos.

Además de la gran velocidad de transferencia de datos entre el computador y la unidad periférica ATARI 1050, comparada con la unidad de cassette, otra importante diferencia radica en la posibilidad de obtener cualquier información dentro del diskette en forma inmediata (random) y no secuencialmente (una detrás de otra) como en el caso del cassette.

Para operar una unidad de diskette ATARI 1050 se requiere de un programa especial, llamado D.O.S. (Disk Operating System), que viene incluido en un diskette con la compra del equipo. Este D.O.S. (versión 2.5) es el encargado de controlar todas las operaciones y funciones relacionadas con la operación de la diskettera, como ser: lectura y grabación de archivos, directorio del contenido del diskette, borrar archivos, copiar archivos de un diskette a otro, duplicar un diskette completamente y en general, todo lo necesario para que su computador ATARI se pueda comunicar con la unidad de diskette ATARI 1050.



También, con la compra del ATARI 1050 COELSA COMPUTACION entrega un diskette de regalo, conteniendo una serie de programas varios, que permiten el mejor conocimiento de operación de este periférico.

Sin lugar a dudas, la adición de una unidad de diskette ATARI 1050 a su configuración computacional le abrirá un mundo de nuevas y sofisticadas posibilidades, que no son factibles de lograr con una unidad de cassette, especialmente en la utilización de programas envasados, como son: planillas electrónicas, procesadores de texto, bases de datos, etc., que transforman el computador en una poderosa herramienta de apoyo personal y profesional.

CARACTERISTICAS

- Interfase serial directa al computador
- Razón de transferencia: 19,2 Kbits/seg.
- Tiempo de acceso promedio: 74 miliseg.
- Tiempo de acceso máximo: 236 miliseg.
- Lógica de control incorporada
- Capacidad de almacenamiento: 127 Kbytes
- Chequeo de error escritura (CRC)

Videomanía







Esta columna está destinada a tí, fanático de los juegos, para mantenerte informado sobre las novedades y preferencias de los juegos disponibles en el mercado internacional ATARI.

Una de las actividades propias de la columna será la de mantener un **ranking** de los 12 juegos que acaparan la atención de los usuarios chilenos del microcomputador ATARI.

¿Cómo puedes participar?

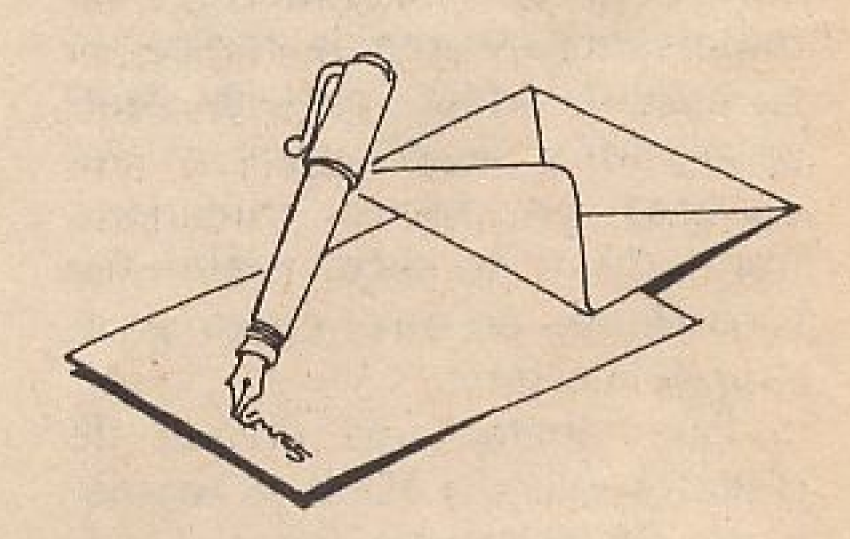
Enviando una carta a:

VIDEOMANIA
MUNDOATARI
Casilla 458, Correo Ñuñoa
Santiago

Al reverso anota el nombre del juego que más te motiva a manipular con tu joystick, tu nombre, dirección, número de socio y número de suscriptor.

Todos los meses publicaremos la lista de los 12 juegos más solicitados.

Y además se sortearán 30 premios entre los que envíen cartas a nuestra sección.



OTRA FORMA DE PARTICIPAR . . .

Otra modalidad de participación en la columna de MUNDOATARI es la de preparar una breve descripción de tu juego favorito, para publicarla en la revista.

La pauta de evaluación será: 1. Nuevos juegos de producción actual en el mercado internacional, 2. Juegos antiguos que por su características no pierden vigencia, y 3. La calidad de la descripción, redacción y presentación del escrito.

***NOTA:** para efectos de ilustración necesitamos una copia del programa que será devuelta posteriormente.*



World Championship Karate

Nombre : *World Championship Karate*
Productor : *Epyx*
Computador : *ATARI*
Memoria : *48 Kb*

CALIFICACION (nota de 1 a 7)	
Gráficos	: 6.0
Sonido	: 6.0
Animación	: 7.0
Dificultad	: 7.0
Originalidad	: 6.5
Promedio	**: 6.5**



Para quienes han jugado el popular juego de video Karate Champ resultará agradable encontrar una versión similar para el computador ATARI. Se trata de **World Championship Karate,** elaborado por la firma norteamericana Epyx.

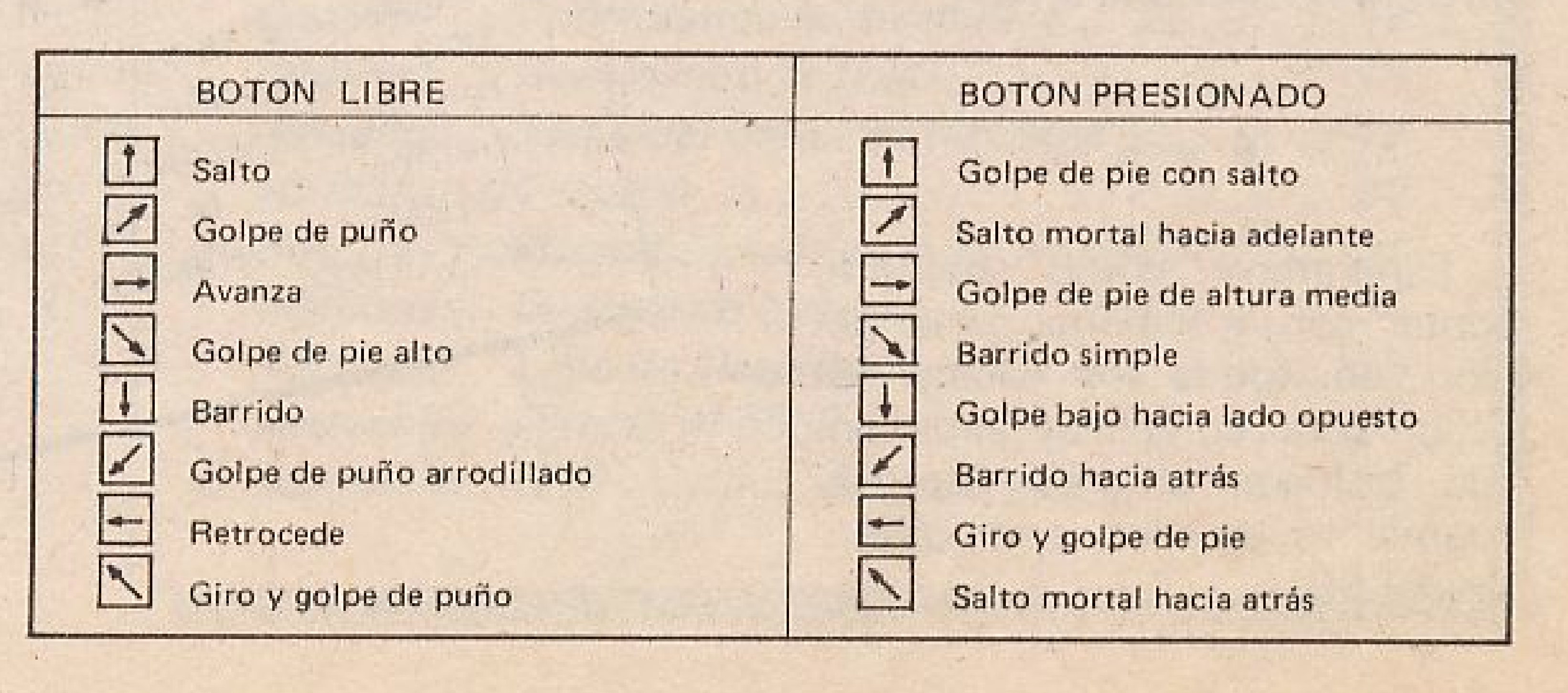
Consiste en un campeonato internacional de Karate en el que intervienen hasta 2 jugadores. Tu tarea será la de enfrentarte a un karateka vestido de rojo, controlado por el computador o por el segundo jugador. A medida que vas venciendo a tus oponentes va cambiando el paisaje y aumentando el nivel de dificultad.

El campeonato se desarrolla en 8 escenarios, entre los cuales figuran: Sydney, Nueva York, Grecia, Egipto, Sao Paulo, monte Kilimanjaro, etc.

Los 8 movimientos del joystick son combinados también con el uso del botón rojo, con lo que se obtiene 16 movimientos.

Cada combate dura 30 segundos y es interrumpido cuando uno de los karatekas golpea al otro. En ese momento el árbitro determina el valor del golpe, que puede ser uno o medio punto. Gana el primero que alcance 2 puntos.

En resumen, se trata de un juego de acción exitante y entretenido. Lo más destacable es la belleza y precisión de sus múltiples golpes. Excelente animación y buenos gráficos.



BOTON LIBRE		BOTON PRESIONADO	
↑	Salto	↑	Golpe de pie con salto
↗	Golpe de puño	↗	Salto mortal hacia adelante
→	Avanza	→	Golpe de pie de altura media
↘	Golpe de pie alto	↘	Barrido simple
↓	Barrido	↓	Golpe bajo hacia lado opuesto
↙	Golpe de puño arrodillado	↙	Barrido hacia atrás
←	Retrocede	←	Giro y golpe de pie
↖	Giro y golpe de puño	↖	Salto mortal hacia atrás

CLUB ATARI DE CHILE INFORMA

Como informamos en la edición especial de MUNDOATARI No. 0 (abril 1987), COELSA COMPUTACION ha creado el ATARI CLUB DE CHILE, el primero de su especie en nuestro país.

Con la creación de este club, todos los poseedores de computadores ATARI-COELSA, pasarán a ser socios activos del club, desde el momento que remitan a COELSA COMPUTACION el Registro de Garantía, que viene con su equipo, con todos sus antecedentes.

A vuelta de correo, el cliente recibirá su credencial de socio del ATARI CLUB DE CHILE, junto a otras sorpresas preparadas para él. Esta credencial de socio le permitirá tener acceso a todos los eventos que se desarrollarán en forma exclusiva para los miembros del Club.

Es importante hacer notar, que esta credencial de socio, es de uso exclusivo del socio titular, indicado en el Registro de Garantía enviado, y por lo tanto no es transferible a otras personas, debiendo guardarla en un lugar seguro para evitar su pérdida.

Destacamos algunas de las diversas actividades para los miembros del Club, preparadas por COELSA COMPUTACION:

* * *

Esperamos como siempre, contar con la máxima participación de todos los socios del Club, para compartir experiencias y lograr el mejor aprovechamiento de los computadores ATARI.

✱ CONCURSO ATARI SUPER SCORE

La oportunidad de ganar $ 1.000.000 en dinero efectivo o cientos de premios en Personal Stéreos, radiorelojes o minicomponentes con radio AM/FM.
Para participar, sólo debe enviar las Tarjetas de Garantía de los productos ATARI-COELSA indicados en los informativos que se te hacen llegar en forma periódica, a fin de ir acumulando puntos para el gran concurso de fin de año.

✱ CONCURSOS DE SOFTWARE

Se ha programado la realización de dos concursos de desarrollo de software, uno a nivel de usuarios y otro para colegios. De este modo, todos los socios del Club y de los colegios que posean computadores ATARI, que desarrollen un programa computacional ATARI de creación propia y original, podrán participar en este evento. Los ganadores obtendrán interesantes premios por su colaboración.

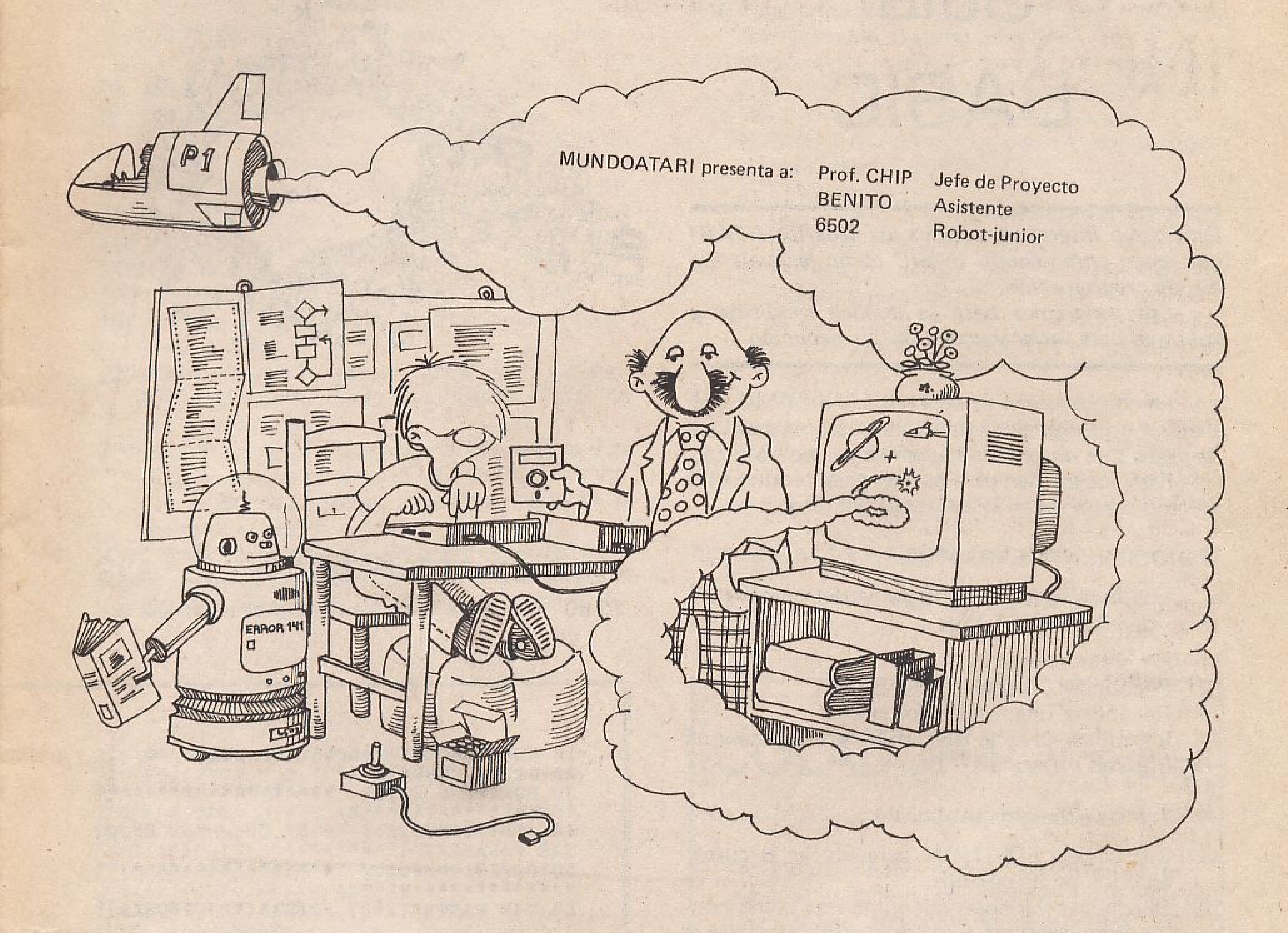
✱ PROMOCION DE PRODUCTOS

A fin de que todos los socios del Club puedan ir incrementando su equipamiento ATARI, se realizarán interesantes promociones de periféricos y programas, a precios especiales para los socios.

✱ SUSCRIPCION A REVISTA MUNDOATARI

Todos los socios del Club tendrán acceso a tomar una suscripción anual a esta revista a precio preferencial, recibiendo mensualmente en su hogar, un ejemplar de la revista con las últimas novedades, noticias, enseñanzas y programación del computador ATARI.

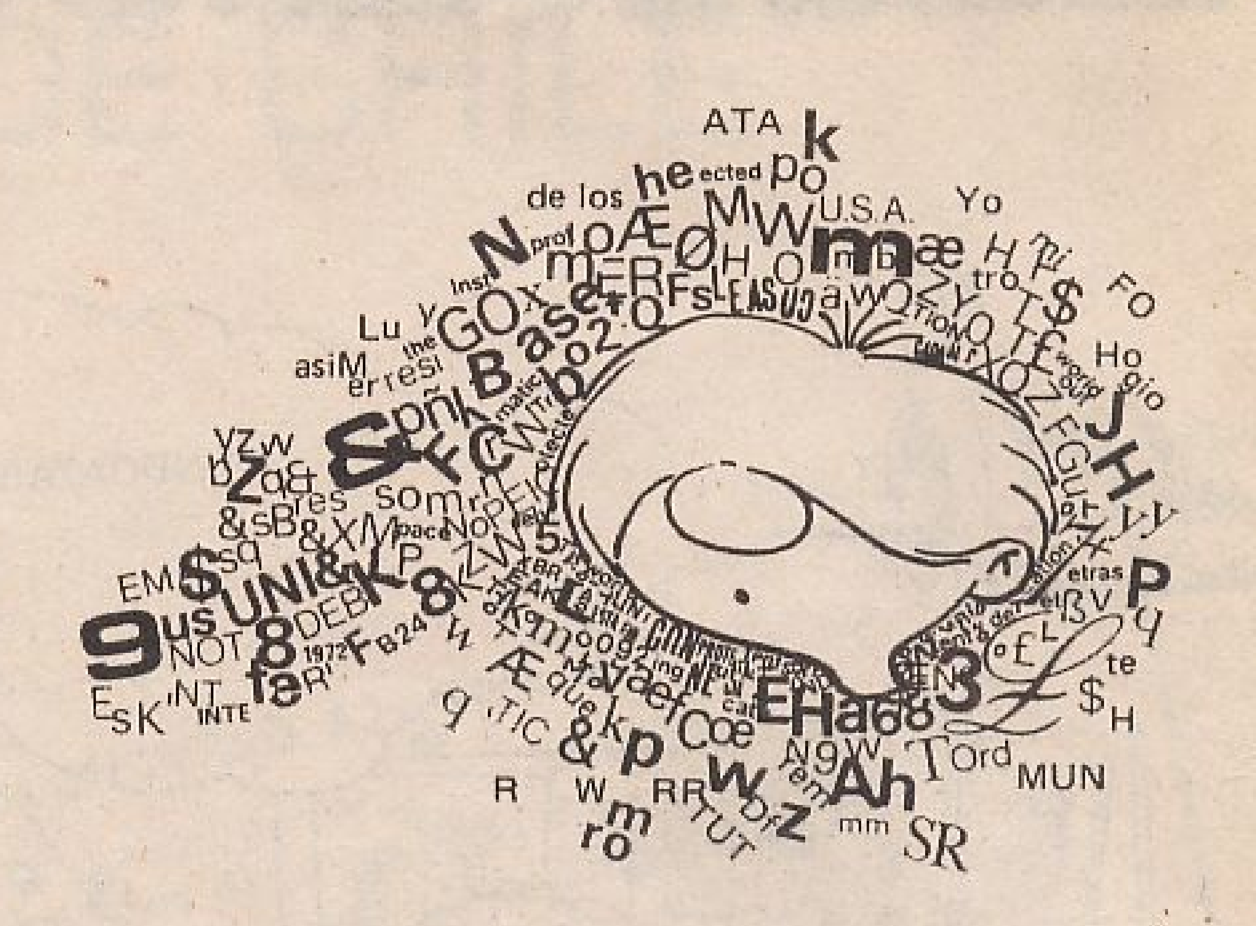
MANEJANDO TU ATARI



El bloque MANEJANDO TU ATARI proporciona a nuestros lectores con sus 9 columnas de continuidad: información, programas y actividades para todo nivel.
Usted tomará conciencia de su estado actual de conocimiento, según el número de columnas en que participe.
Si usted encuentra dificultad en alguna de ellas no desespere, pues al avanzar en otras ganará experiencia y podrá retomarla con el tiempo.

INDICE DEL BLOQUE

Guía BASIC



Este curso interesará a todos los usuarios ATARI que estén aprendiendo BASIC como lenguaje inicial de programación.

Los programas base se pueden modificar a voluntad para variar y acomodar su ejecución.

La continuación de GUIA ATARI BASIC consiste en un juego elemental para buscar la cantidad de veces que una palabra se encuentra en un texto.

Este juego manipula strings para producir los efectos esperados en la estrategia del juego.

CONOCIMIENTOS PREVIOS

Los indicados en el programa del número anterior de MUNDO ATARI:

a) Encender el equipo.
b) Reconocer la presencia del BASIC.
c) Reconocer una línea de programa.
d) Identificar la tecla RETURN para ingresar una línea en memoria.

En este programa se aplicará:

a) Instrucción POSITION para ubicar el cursor en posición de pantalla.
b) Reconocer variables alfanuméricas como conjunto de letras mayúsculas que finalizan con el signo "$".
c) El contenido de variable alfanumérica es considerado por el ATARI BASIC como una secuencia de caracteres.
d) El ATARI BASIC debe informarse acerca de la DIMensión máxima para una variable alfanumérica.
e) Buscar un número al azar con función RND.

ENCONTRAR PALABRAS

Tipee cuidadosamente el siguiente programa. Recuerde que al finalizar una línea debe presionar la tecla RETURN.

Cumplida esta etapa podrá ejecutarlo y desarrollar la habilidad de buscar palabras en un texto de 200 letras.

Usted puede acumular un máximo de 7.000 puntos si acierta en los 7 intentos.

DESARROLLO DEL PROGRAMA

10-50 Formatea la pantalla para el título del juego

```
10 REM JUEGO de encontrar palabras
20 PRINT CHR$(125)
30 POSITION 2,0:? "*******************
*****************";
40 POSITION 2,1:? "*     JUEGO DE ENCON
TAR PALABRAS     *"
50 POSITION 2,2:? "*******************
*****************";
60 DIM PANTA$(200),PABRA$(10),PAUSA$(1
)
70 PANTA$=""
80 CUENTA=4:PUNTOS=0
90 FOR J=0 TO 6
100 PABRA$=""
110 POSITION 0,10:? " DIGITE  una pala
bra de ";CUENTA;" letras"
120 INPUT PABRA$
130 L=LEN(PABRA$)
140 IF L<=CUENTA THEN 190
150 POSITION 0,20:? " INGRESO una pala
bra con ";L;" letras"
160 POSITION 0,21:? " PRESONE una tecl
a y continua"
170 INPUT PAUSA$
180 GOTO 100
190 ? CHR$(125)
195 POSITION 0,10:? "ESPERE FORMATEAND
O EL TEXTO........";
200 FOR K=1 TO 200
210 AZAR=INT(RND(1)*L)+1
220 PANTA$(K,K)=PABRA$(AZAR,AZAR)
230 NEXT K
300 REM PANTALLA DE BUSQUEDA
310 ? CHR$(125):POKE 82,0
```

60 Dimensiona las variables alfanuméricas que utilizará:
PANTA$: Texto total con 200 caracteres
PABRA$: Palabra ingresada para buscar en el texto
PAUSA$: Carácter utilizado para crear una pausa
70 Inicializa el TEXTO como string nulo
80 Inicializa las variables numéricas:
CUENTA : Número de letras de la palabra ingresada (4)
PUNTOS : Cantidad de puntos obtenidos (0)
90 Ciclo para 7 intentos. Variable CONTROL J
100 Inicializa la palabra como string nulo
110-180 Ingresa la palabra a buscar
190 Carácter específico que limpia la pantalla
195 Mensaje indicando una pausa para que ATARI cree un TEXTO
200-230 Ciclo para crear un string en la variable PANTA$ con los caracteres elegidos al azar de la variable PABRA$
300-370 Presenta una pantalla con la palabra y el texto para que el usuario interactúe buscando el número de veces que la palabra ingresada se repite en el texto
390 Ingresa el número en la variable numérica NUM
405 Inicializa la variable REPET que contendrá el número de repeticiones que buscará el ATARI (0)
410-440 Rutina de manejo de string para buscar el número de palabras PABRA$ en PANTA$.

Detalle:
410 Establece ciclo entre los límites 1 y 200 que corresponden al string PANTA$
420 Mediante el uso de índices para el string PANTA$ elige el que corresponde a la primera palabra, en función del largo del string.
Condiciona la dirección según la comparación con la palabra PABRA$:
a) Si son iguales:
— Incrementa la variable contadora del número de repeticiones REPET
— Incrementa la variable K en el largo de palabra menos uno.
b) Si son diferentes:
— Incrementa en uno (NEXT) la variable control del ciclo (K)
450 Compara el valor calculado por el ATARI y el usuario. Si son iguales incrementa en 1.000 la variable contadora de puntos
500 Limpia la pantalla e incrementa la variable número de letras de palabra (variable CUENTA).

ACTIVIDADES DE SUPERACION

1. Permitir que participen dos jugadores y el ATARI.

```
315 POSITION 12,1:? PABRA$
320 POSITION 0,2:? "******************
************************";
330 POSITION 0,4:? PANTA$
340 POSITION 0,10:? "*****************
*************************";
350 POSITION 0,14:? "BUSQUE..........
.........................";
360 POSITION 0,15:? " el numero de pal
abras  ";PABRA$
370 POSITION 0,16:? " que aparecen en
el TEXTO.-"
380 POSITION 0,18:? " DIGITE el numero
 "
390 POSITION 2,19:INPUT NUM
400 REM BUSQUEDA
405 REPET=0
410 FOR K=1 TO 200
420 TRAP 450:IF PANTA$(K,K+L-1)<>PABRA
$ THEN 440
430 REPET=REPET+1:K=K+L-1
440 NEXT K
450 IF NUM=REPET THEN PUNTOS=PUNTOS+10
00:GOTO 500
460 ? CHR$(125)
470 POSITION 0,10:? "PERDIO........"
480 POSITION 10,11:? "eran ";REPET;"ve
ces ";
490 POSITION 0,20:? "PRESIONE una tecl
a y continua"
495 INPUT PAUSA$
500 ? CHR$(125):CUENTA=CUENTA+1
510 NEXT J
520 ? CHR$(125)
900 END
```

COLABORACIONES

MUNDOATARI recibe artículos, programas y columnas de continuidad para ser editadas en los próximos números.

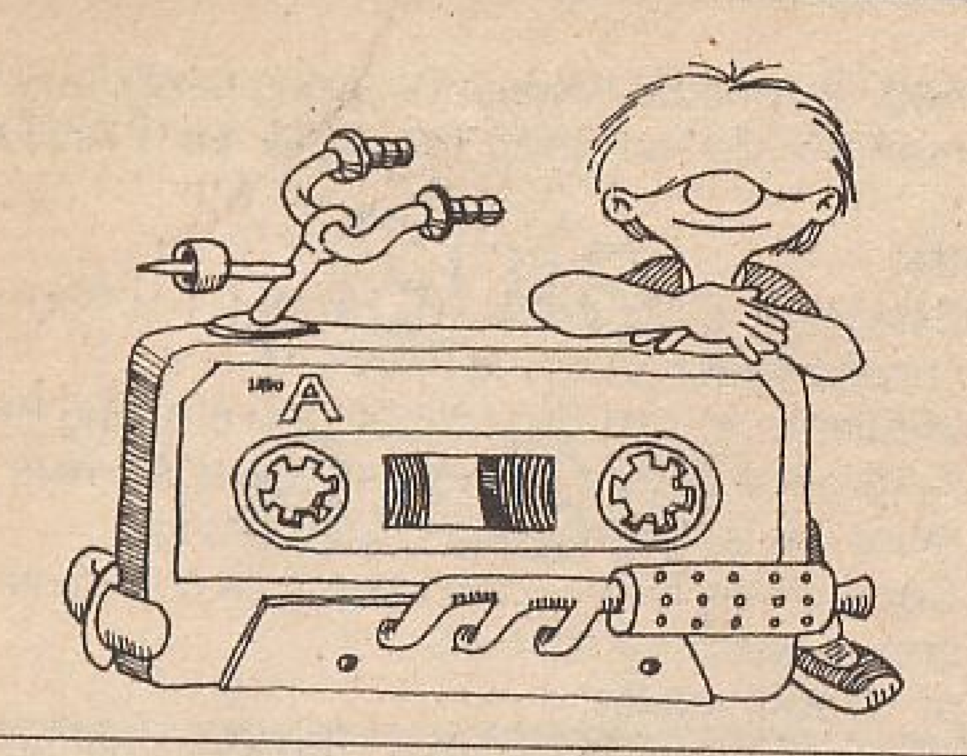
Estas columnas serán evaluadas y analizadas de acuerdo a nuestra línea editorial.

Previo a la publicación de un aporte, la revista enviará el dinero correspondiente a su pago, constituyéndose por este hecho en propiedad de la revista.

Las contribuciones no aceptadas por la revista serán devueltas a sus autores.

Los artículos deben ser escritos a máquina o en procesador de textos. Los programas, grabados en casete o diskette y listados además en impresora.

La publicación de cualquier material en las páginas de MUNDOATARI estará supeditada a las disponibilidades de espacio o a las necesidades de la edición.



Controlando la casetera

SOLUCION A NUMERO ANTERIOR

La respuesta correcta a la pregunta planteada en el número anterior era: LIST "C:" / ENTER "C:"

En efecto, estas instrucciones permiten grabar un programa fraccionado sin riesgo de borrar otras líneas que existan en memoria.

Sin embargo, es conveniente hacer algunos alcances:

1. *Se debe asignar claramente el rango del tipeo. El usuario A digita desde línea 1 a 255 y el usuario B de la 256 al final.*
2. *Al juntar ambas partes con ENTER "C:" se debe tipear previamente NEW o encender de nuevo el computador. Recuerde que cualquier línea remanente de otro programa que no sea reemplazada por otra de igual número será incorporada al programa final e interferirá en su ejecución.*

En el número anterior tratamos los comandos usados para grabar y cargar programas mediante la casetera. También comentamos la estructura de un archivo en cinta magnética. Estos conceptos estarán presentes continuamente en los próximos números de esta columna.

Hoy iniciaremos la técnica para manejar archivos de datós en casete.

B) ARCHIVOS DE DATOS

El casete permite manejar archivos de datos, esto es, grabar información, mantenerla almacenada en ese medio magnético y posteriormente recuperarla.

Los archivos en casete comparados con el diskette tienen algunas limitaciones que mencionaremos a título informativo:

- son más lentos
- los datos deben ser accesados secuencialmente, o sea, leídos en orden y de principio a fin
- no se puede agregar ni borrar información, salvo que se regrabe completamente.

¿Cuánta información cabe en un casete?. En las especificaciones de la casetera dadas en el número anterior de MUNDOATARI se menciona una capacidad aproximada de 100 Kb. en un casete de 60 minutos. Obviamente depende esto de la cantidad de archivos grabados en ella. Recordemos que cada archivo tiene su propio tiempo de inicio y de finalización (EOF), los cuales inciden en el espacio útil de la cinta.

La unidad de archivo en casete es un bloque de 128 bytes. Un archivo menor a esa cantidad de bytes ocupa al menos un bloque, un archivo de mayor extensión ocupará los bloques que sea necesario para contener toda la información. La estructura de un archivador de correspondencia ilustra mejor esta forma de archivo: la unidad es la hoja de carta: Hay cartas largas que se extienden a varias hojas y hay cartas muy breves de una hoja, pero no se encuentran fracciones de hojas.

El ATARI BASIC maneja archivos de datos con dos instrucciones de grabación y carga respectivamente:

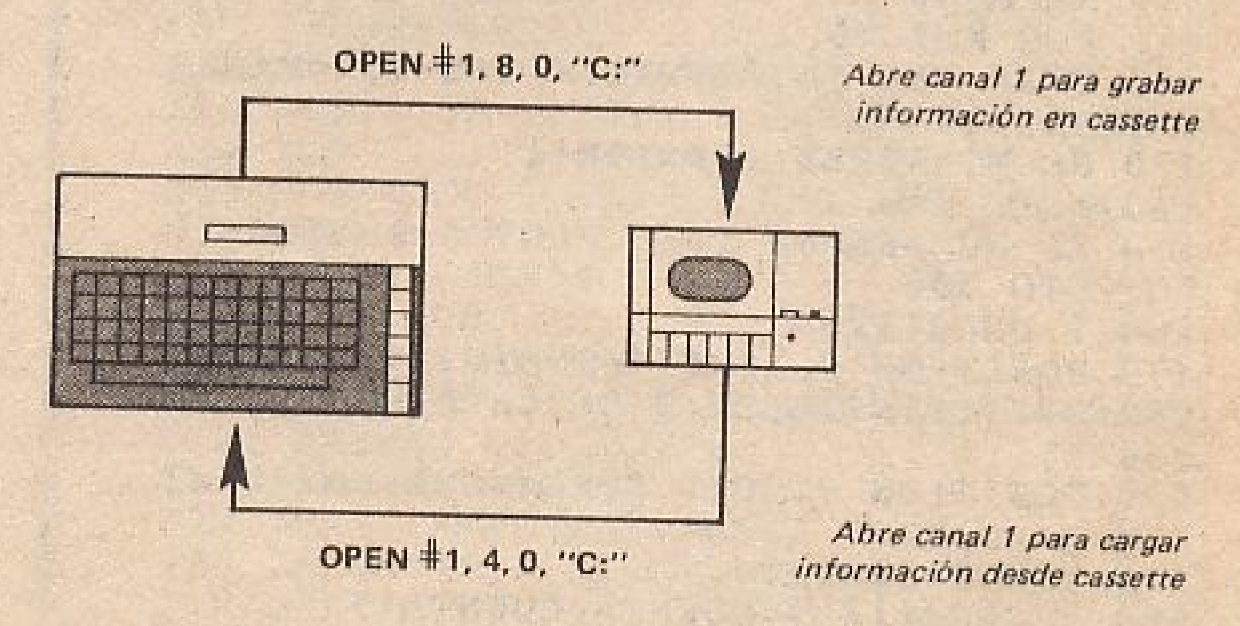
- **PRINT #/ INPUT #**
- **PUT #/ GET #**

En este número veremos las dos primeras.

ABRIR Y CERRAR CANALES

A diferencia de la técnica usada en la grabación de programas, para grabar o cargar archivos de datos se debe abrir o cerrar **canales de comunicación** entre el computador y la casetera. Nosotros hacemos lo mismo cuando queremos hablar por teléfono con una persona determinada: marcamos un número específico y así obtenemos una vía de comunicación con ella.

En la práctica se usa 2 instrucciones básicas que se ilustran en el ejemplo siguiente:



El número 1 que figura en OPEN #1 se refiere al número de canal de comunicación, que puede ser indistintamente 1, 2, 3, 4 ó 5. Aún cuando hay 8 canales en total (0-7) BASIC ocupa los canales

0, 6 y 7 para funciones específicas.

En el segundo parámetro se usa los números 8 para grabar en casete ó 4 para cargar información desde la casetera.

En el tercer parámetro consideraremos el número 0 como velocidad standard.

La letra "C:" (con comillas y dos puntos) se refiere a la designación abreviada de **casetera** como periférico.

CREACION DE ARCHIVOS CON PRINT #/ INPUT

Estas instrucciones sirven para grabar o cargar cualquier información proporcionada en forma textual o en una variable alfanumérica (string)

```
10 OPEN #1, 8, 0, "C:"
20 PRINT #1; "YO LEO MUNDOATARI"
30 CLOSE #1
```

o bien:

```
10 DIM A$(20): A$ = "PRUEBA 1234567890"
20 PRINT #1; A$
30 CLOSE #1
```

En ambos casos el computador emite 2 sonidos BEEP y queda a la espera de que pongamos la cinta al comienzo y presionemos RECORD/PLAY en la casetera. Al momento de digitar cualquier tecla en el computador se inicia el proceso de grabación.

La línea 30 cierra el canal, lo que hace poner en la cinta la marca de fin de archivo (EOF), que indicará en el futuro donde termina el archivo. El hecho de no cerrar el canal significa que arriesgamos perder los últimos datos de la grabación.

Digite el siguiente listado para practicar lo aprendido:

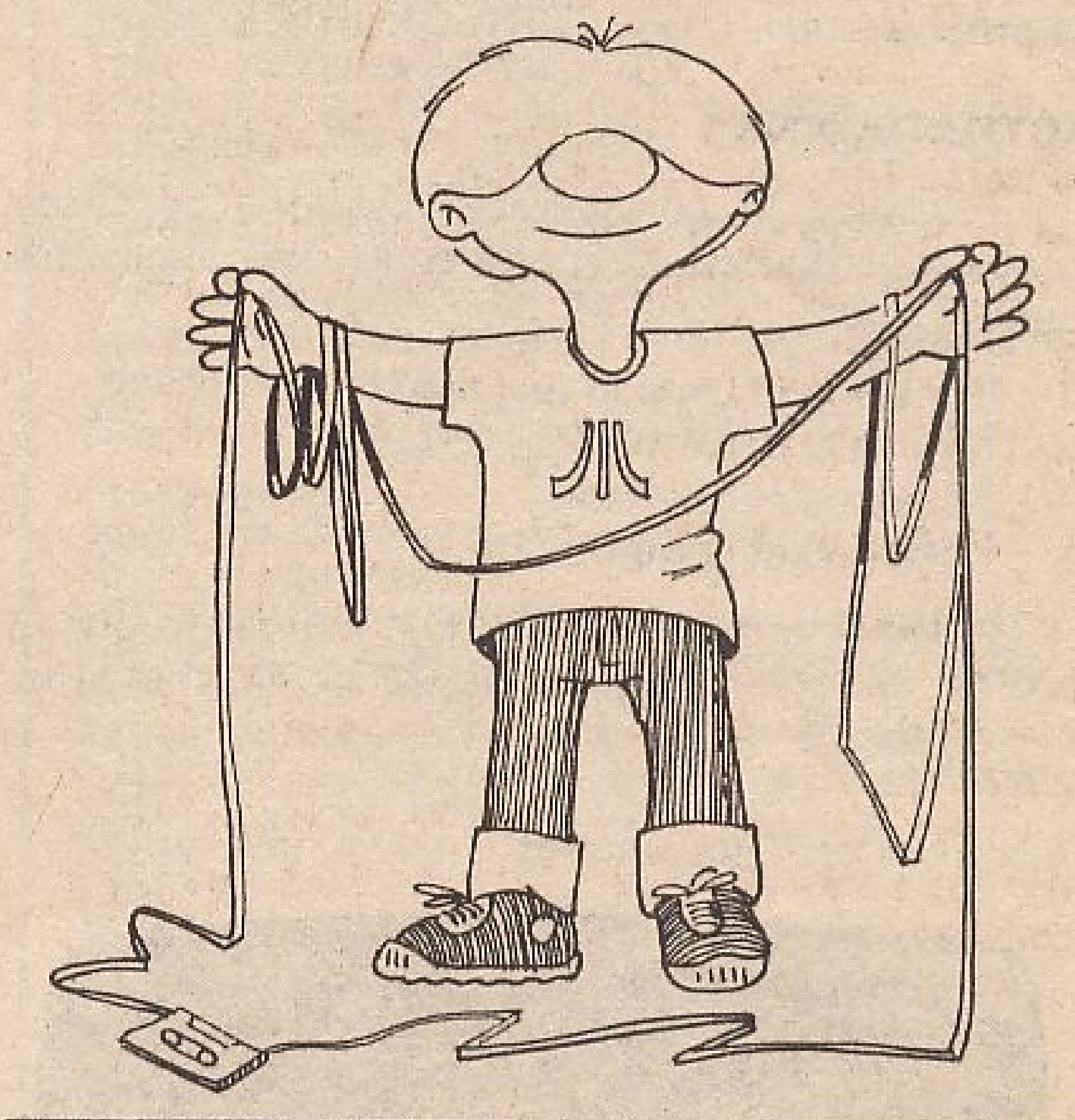
```
10 GRAPHICS 0
15 DIM ORI$(40),COPIA$(40)
20 PRINT "Escriba mensaje de 1 linea":
PRINT
25 INPUT ORI$
30 PRINT "Prepare cinta. Presione RECO
RD+PLAY"
35 PRINT "Digite una tecla para empeza
r"
40 OPEN #1,8,0,"C:"
45 PRINT #1;ORI$
50 CLOSE #1
55 PRINT :PRINT :PRINT "Digite CONT+RE
TURN para seguir"
60 STOP
65 REM *CARGANDO*
70 PRINT CHR$(125)
75 PRINT "Prepare cinta. Presione PLAY
"
80 PRINT "Digite una tecla para empeza
r":PRINT :PRINT
85 OPEN #2,4,0,"C:"
90 INPUT #2;COPIA$
95 CLOSE #2
100 PRINT "COPIA$= ";:PRINT COPIA$
105 PRINT "ORI$= ";:PRINT ORI$
```

Al ejecutar este programa efectuamos la prueba de grabar una línea que usted escribe libremente en la variable ORI$ y posteriormente la recuperamos en la variable COPIA$ para comparar finalmente el contenido de ambas variables.

El listado incluye mensajes orientadores para informar al usuario de lo que debe hacer, de modo de no sorprenderlo con signos de interrogación o sonidos imprevistos.

Comentario del listado:

- **30-50** Programa para grabar ORI$ al casete
- **15** Dimensionamiento de las variables ORI$ y COPIA$
- **40** Apertura del canal 1 para grabación
- **45** Grabación de ORI$
- **50** Cierre del canal, que genera un EOF
- **60** Detención momentanea para pasar al próximo paso. El programa continúa con la instrucción CONT
- **75-95** Programa para cargar archivo desde la casetera a la variable COPIA$
- **75-80** Edición de instrucciones
- **85** Apertura de canal 2 para cargar (el número puede ser cualquiera entre 1 y 5 siempre que esté libre. El canal 1 en este caso ya estaba desocupado)
- **90** Carga del archivo a COPIA$
- **100-105** Edición de las 2 variables en pantalla para comparar contenidos.



PROBLEMA PARA EL PROXIMO NUMERO

1. Modifique el listado para grabar y cargar un mensaje de 3 líneas de largo.
2. Modifique el listado para grabar y cargar 2 variables consecutivas de 40 caracteres cada una. Por ejemplo: NOMBRE$, DIRECCION$.

(Solución en el próximo número)

Dominando la 1050



La continuación de esta columna desarrollará actividades prácticas para interactuar con los archivos almacenados en el directorio utilizando el MENU del DOS 2.5.

El primer artículo menciona el Menú para inicializar un diskette con la opción Formateo.

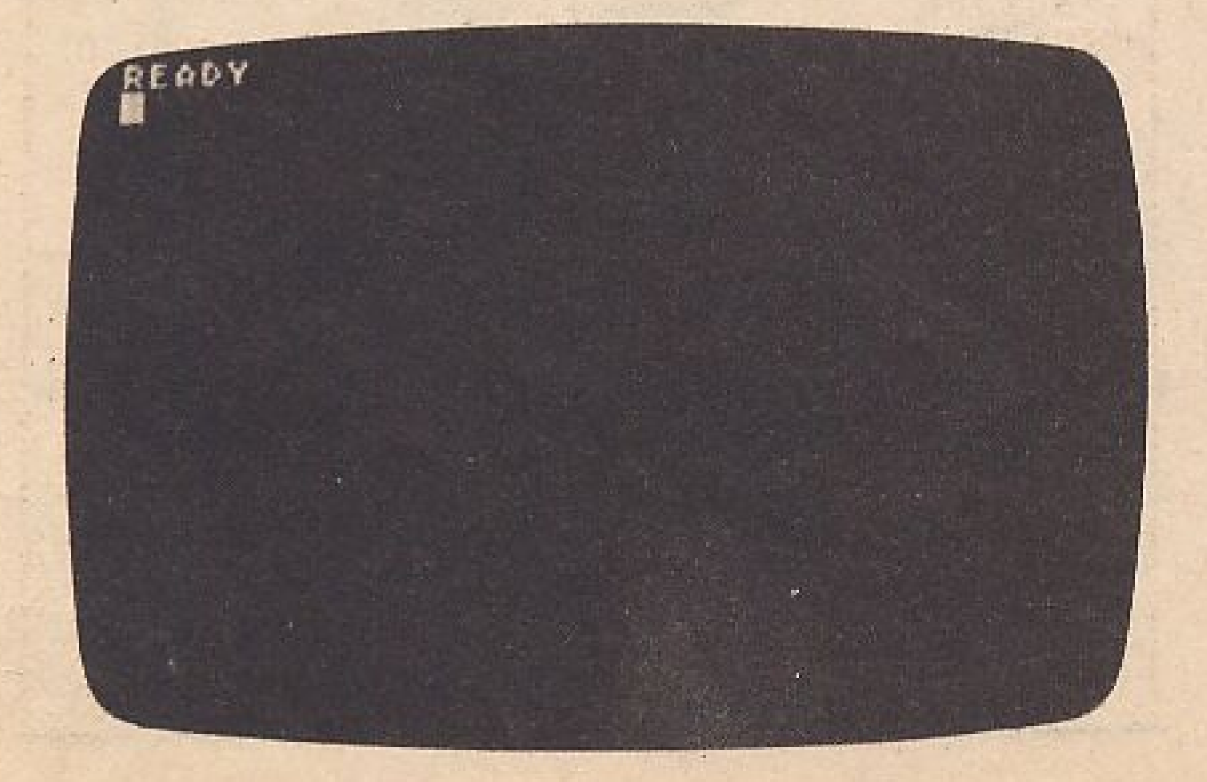
Se describió en extenso la constitución del diskette como las estructuras de almacenamiento de nombres de archivo. (DIRECTORIO).

ACTIVIDAD No.1

Inicialice la configuración Unidad de Disco y microcomputador cumpliendo las siguientes etapas:

a) Encienda la Unidad de Disco.
b) Inserte el diskette DOS 2.5.
c) Encienda el televisor.
d) Encienda el computador.

Aparece el mensaje READY indicando que el BASIC se encuentra incorporado en memoria.



ACTIVIDAD No. 2

Inserte un diskette formateado, ver artículo del número anterior de MUNDO ATARI.

Digite el siguiente programa:

```
10 REM PROGRAMA No. 1
20 FOR J=0 TO 255 STEP 16
30 POKE 710, J
40 NEXT J
50 END
```

Ejecute el programa con instrucción RUN.

Observe los cambios de color de la pantalla. La instrucción POKE modifica el valor para la localización que se indique, en este caso es la 710.

ACTIVIDAD No. 3

Salve este programa en el diskette.

La instrucción SAVE del ATARI BASIC permite almacenar el programa residente en memoria en el diskette que se encuentra en el interior de la Unidad de Disco.

Esta instrucción requiere de caracteres que en su conjunto identifiquen con un nombre a este programa.

ACTIVIDAD No. 4

El nombre es COLOR.

Agregue los caracteres correspondientes para que se ejecute la acción de SALVAR (Grabar).

Recurra a:

a) Número anterior de MUNDO ATARI.
b) Manual de la Unidad de Disco.

El nombre debería ser:

SAVE "D: COLOR.BAS"

y presionar la tecla RETURN.

Con esta operación inmediatamente se enciende la luz roja de la Unidad de Disco indicando que la información se encuentra en proceso de transferencia.

ACTIVIDAD No. 5

¿Cómo ver este nombre en el Directorio?

Para cumplir con estas acciones, se necesita

del MENU del DOS 2.5.

Este menú se llama con la palabra DOS.

PRECAUCION: Al realizar este llamado es imprescindible que en el directorio del disco presente en la disketera se encuentren los archivos DUP.SYS y DOS.SYS.

Si el diskette de trabajo no lo contiene, cambie por el diskette del DOS 2.5.

Si el llamado es correcto se coloca en acción nuevamente la Unidad de Disco y en la pantalla se despliega el menú de trabajo que sigue:



Si cambió al DOS 2.5 Master, inserte nuevamente el disco de trabajo en que salvó (grabó) el programa COLOR.

Seleccione la opción A y presione la tecla RETURN dos veces y ... ahí está el archivo que usted grabó.

ACTIVIDAD No. 6

Regresemos al programa:

Para ello, presione RETURN y aparece nuevamente el MENU.

La opción B nos permite dejar este menú y retornar al lenguaje ATARI BASIC.

El mensaje identificador READY se encuentra con nosotros.

ACTIVIDAD No. 7

Ejecutemos nuevamente nuestro programa COLOR.BAS. Digitamos RUN, y .... no pasa NADA. Es decir, el programa se nos fue. ¿Por qué?. En los ATARI 800 si no se recurre a crear un archivo de almacenamiento transitorio llamado MEM SAV la información del programa BASIC residente en memoria desaparece y es ocupado por el programa del DOS.

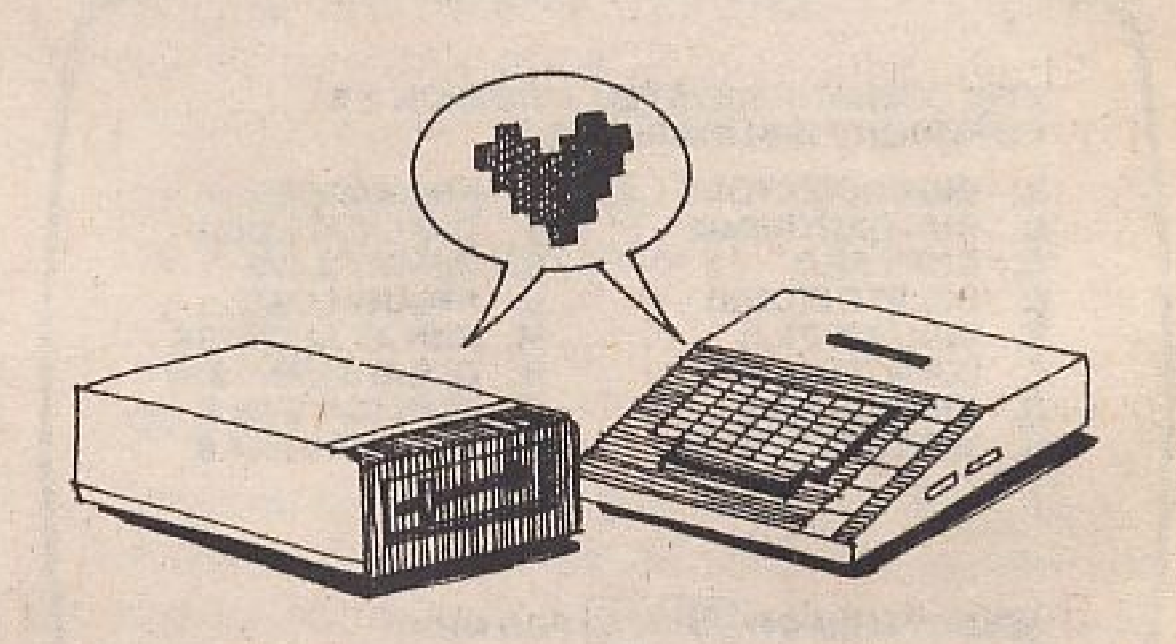
COMENTARIO: La situación es diferente para el ATARI 130 XE, en que si el diskette inicial de partida contiene el RAMDISK.COM el banco secundario permite contener simultáneamente ambos programas.

El programa listado a continuación, llamado MINIMENU es un programa BASIC que puede incorporar las principales tareas del DOS directamente sin perder el programa residente en memoria.

```
30000 REM MINIDOS 2.5 version -1 para
      MUNDOATARI MAYO 1987
30001 REM para usar en sus programas
      borre lineas #
30030 DIM ARCHI$(30),ANS$(1),ARCH1$(12
),B$(40)
30035 B$(1)=" ":B$(40)=" ":B$(2)=B$
30040 GRAPHICS 0:POKE 16,112:POKE 5377
4,112:POKE 752,1
30045 POSITION 0,0:? "            FUNCI
ONES  DOS 2.5              ";
30048 POSITION 2,1:? "SES":POSITION 0,
2:? "SISTEMA"
30050 POSITION 6,4:? " A.  DIRECTORIO"
30052 POSITION 6,6:? " B.  LENGUAGE BA
SIC"
30054 POSITION 6,8:? " D.  BORRA ARCHI
VO(S)"
30056 POSITION 6,10:? " E.  CAMBIA NOM
BRE"
30058 POSITION 6,12:? " F.  PROTEGE AR
CHIVOS"
30060 POSITION 6,14:? " G.  DESPROTEGE
 ARCHIVOS"
30062 POSITION 6,16:? " I.  FORMATEA D
OBLE DENSIDAD"
30070 POSITION 0,18:? "       Z.  fina
liza  DEMOSTRATIVO # 1   ";
30080 POSITION 0,20:? " DIGITA   letra
  corresponda a FUNCION  ";:POKE 752,0
30082 INPUT ANS$
30084 IF ANS$="Z" THEN GOTO FIN
30090 IF ANS$<"A" OR ANS$>"I" OR ANS$=
" " OR ANS$="H" OR ANS$="C" THEN 30040
30100 KEY=ASC(ANS$)-64:IF KEY=2 THEN ?
 CHR$(125):END
30110 ? CHR$(125):POSITION 0,0:? "
         DIRECTORIO                  ";
:TRAP 30900
30120 TRAP 30150:CLOSE #2:OPEN #2,6,0,
"D:*.*":CUENTA=2
30130 CUENTA=CUENTA+1:INPUT #2,ARCHI$:
POSITION 3,CUENTA:? ARCHI$:IF ASC(ARCH
I$(1,1))>47 THEN 30150
30140 CUENTA=CUENTA+1:INPUT #2,ARCHI$:
POSITION 3,CUENTA:? ARCHI$;:IF ASC(ARC
HI$(1,1))<47 THEN 30130
30142 FOR TIME=1 TO 500:NEXT TIME:GOTO
 30130
30150 CLOSE #2:ON KEY GOTO 30160,30160
,30160,30165,30230,30340,30400,30160,3
0460
30160 POSITION 0,20:? "        PRESIONA
  RETURN  para  MENU      ";:INPUT ANS$
:GOTO 30040
30165 POSITION 0,2:? " BORRA ARCHIVO
 RETURN     finaliza.... ";
30170 POSITION 0,20:? " DIGITA  nombre
  de ARCHIVO que  borra  ";:INPUT ARCH
1$
30180 IF ARCH1$="" THEN 30040
30190 POSITION 0,22:? " BORRA    ";ARCH
1$;" S/N ? ";:INPUT ANS$
30200 IF ANS$="S" THEN 30210
```

```
30205 POSITION 0,21:? B$;:POSITION 0,2
2:? B$;:POSITION 0,23:? B$(1,38);:GOTO
 30165
30210 L=LEN(ARCH1$):ARCHI$(3,L+2)=ARCH
1$(1,L):ARCHI$(1,2)="D:":ARCHI$=ARCHI$
(1,L+2)
30220 XIO 33,#2,0,0,ARCHI$:GOTO 30110
30230 POSITION 0,2:? " CAMBIA NOMBRE
 RETURN     finaliza.... ";
30235 POSITION 0,20:? " DIGITA  nombre
  de ARCHIVO que   cambia ";:INPUT ARCH
1$
30240 IF ARCH1$="" THEN 30040
30250 POSITION 0,22:? " CAMBIA   ";ARCH
1$;" S/N ? ";:INPUT ANS$
30252 IF ANS$="S" THEN 30270
30255 POSITION 0,21:? B$;:POSITION 0,2
2:? B$;:POSITION 0,23:? B$(1,38);:GOTO
 30230
30270 L=LEN(ARCH1$):ARCHI$(3,L+2)=ARCH
1$(1,L):ARCHI$(1,2)="D:":ARCHI$=ARCHI$
(1,L+2)
30280 POSITION 0,21:? B$;:POSITION 0,2
2:? B$;:POSITION 0,23:? B$(1,38);
30285 POSITION 0,20:? " DIGITA  NUEVO
 nombre  de  ARCHIVO       ";:INPUT ARCH
1$
30290 IF ARCH1$="" THEN 30040
30300 POSITION 0,22:? " CAMBIA   ";ARCH
1$;" S/N ? ";:INPUT ANS$
30302 IF ANS$="S" THEN 30320
30306 POSITION 0,21:? B$;:POSITION 0,2
2:? B$;:POSITION 0,23:? B$(1,38);:GOTO
 30230
30320 L=LEN(ARCHI$):ARCHI$(L+1)=",":AR
CHI$(L+2)=ARCH1$
30330 XIO 32,#2,0,0,ARCHI$:GOTO 30110
30340 POSITION 0,2:? " PROTEGE ARCHIVO
   RETURN     finaliza.. ";
30345 POSITION 0,20:? " DIGITA  nombre
   ARCHIVO que   protege   ";:INPUT ARCH
1$
30350 IF ARCH1$="" THEN 30040
30360 POSITION 0,22:? " PROTEGE   ";ARC
H1$;" S/N ? ";:INPUT ANS$
30362 IF ANS$="S" THEN 30380
30370 POSITION 0,21:? B$;:POSITION 0,2
2:? B$;:POSITION 0,23:? B$(1,38);:GOTO
 30340
30380 L=LEN(ARCH1$):ARCHI$(3,L+2)=ARCH
1$(1,L):ARCHI$(1,2)="D:":ARCHI$=ARCHI$
(1,L+2)
30390 XIO 35,#2,0,0,ARCHI$:GOTO 30110
30400 POSITION 0,2:? "DESPROTEGE ARCHI
VO  RETURN     finaliza.. ";
30405 POSITION 0,20:? " DIGITA  nombre
 ARCHIVO que desprotege  ";:INPUT ARCH
1$
30410 IF ARCH1$="" THEN 30040
30420 POSITION 0,22:? "DESPROTEGE   ";A
RCH1$;" S/N ? ";:INPUT ANS$
30422 IF ANS$="S" THEN 30440
30430 POSITION 0,21:? B$;:POSITION 0,2
2:? B$;:POSITION 0,23:? B$(1,38);:GOTO
 30400
30440 L=LEN(ARCH1$):ARCHI$(3,L+2)=ARCH
1$(1,L):ARCHI$(1,2)="D:":ARCHI$=ARCHI$
(1,L+2)
30450 XIO 36,#2,0,0,ARCHI$:GOTO 30110
30460 POSITION 0,2:? "FORMATEA DOBLE D
ENS #  0     finaliza.. ";
30470 IF ARCH1$="0" THEN 30040
30475 TRAP 30510:OPEN #2,6,0,"D:*.*"
```

```
30480 CLOSE #2:POSITION 0.20:? "      E
XISTEN  ARCHIVOS  EN DISKETTE      ";
30485 POSITION 0,21:? " 1 FORMATEA  2
DIRECTORIO .DIGITA #(1-2)";
30488 CLOSE #1:OPEN #1,4,0,"K:":GET #1
,A:NUMERO=A-48:IF NUMERO=2 THEN 30100
30490 IF NUMERO<>1 THEN GOTO 30480
30500 XIO 254,#2,0,0,"D1:"
30900 POSITION 0,20:? "          DIREC
TORIO    VACIO             ";:FOR TIME=1
 TO 1000:NEXT TIME:GOTO 30040
```



ACTIVIDAD No. 8

La actividad 4 nos salva de la posibilidad de pérdida de la información (programa).

La instrucción LOAD del ATARI BASIC permite trasladar información desde la memoria magnética (diskette) hacia la memoria del computador.

El formato es: **LOAD "D: NOMBRE.EXT"**

Digite la instrucción que recupera el programa COLOR.BAS y presione la tecla RETURN

El texto digitado es:

LOAD "D: COLOR.BAS"

Ejecute el programa y observe que reside nuevamente en la memoria.

Si usted es un nuevo usuario de la Unidad de Disco comprobará que la velocidad de transferencia de la información bien vale la diferencia de su precio.

ACTIVIDAD No. 9

La Unidad de Disco permite reconocer los errores operacionales.

Digite: **LOAD"D: COLOR"** y presione RETURN ¿Qué mensaje envía su ATARI?

ERROR 170

Esto significa que este archivo no se encuentra en el directorio del diskette.

Saque el diskette de la Unidad de Disco y digite: **LOAD"D:COLOR. BAS"**

¿Qué mensaje edita su ATARI ahora? ●

MEMORIX es la adaptación a las potencialidades del computador ATARI de un simpático juego conocido como Simon.

El juego tiene como objetivo ejercitar la habilidad para memorizar una secuencia de sonidos y colores, cuyo orden varía en cada juego.

Los sonidos posibles en una vuelta son cuatro (do, mi, sol, do) y los colores correspondientes son también cuatro (rojo, celeste, amarillo y verde).

El juego puede aumentar el nivel de dificultad hasta llegar a una secuencia de 17 sonidos.

El desarrollo del juego es siempre el mismo: empieza con un sonido y luego va sumando otro hasta llegar al máximo elegido. El jugador debe recordar y reproducir correctamente el modelo presentado y luego digitar en el computador utilizando las teclas:

↑ ↓
← →

En la última vuelta el computador reconoce el triunfo. En caso contrario corrige el error y vuelve al menú para iniciar un nuevo juego. El jugador pierde igualmente si su tiempo de respuesta es mayor de 4 segundos en cada digitación.

Comandos del menú:

Se usan las siguientes teclas:

START: para iniciar el juego
OPTION: aumenta el nivel de dificultad en uno cada vez. Rango de 1 a 9
SELECT: selecciona las dos opciones de juego disponibles: standard y competencia. La primera opción corresponde al juego descrito hasta ahora. Competencia es una modalidad donde los ingresos de sonido son efectuados por el usuario, quien debe repetir la secuencia y aumentarla en uno cada vez.

Comentario del listado:

1. Los 4 colores utilizan el diseño de player, que se encienden o apagan (líneas 310-480)
2. La pantalla tiene una Lista de Despliegue modificada, combinando líneas de modo 1 y 5 (510-550)
3. En la variable A$ se almacena la clave del juego (1020-1050)
4. Las demás rutinas aparecen comentadas en el listado con REM.

¡Hasta el próximo número!

JUEGO DEL MES:

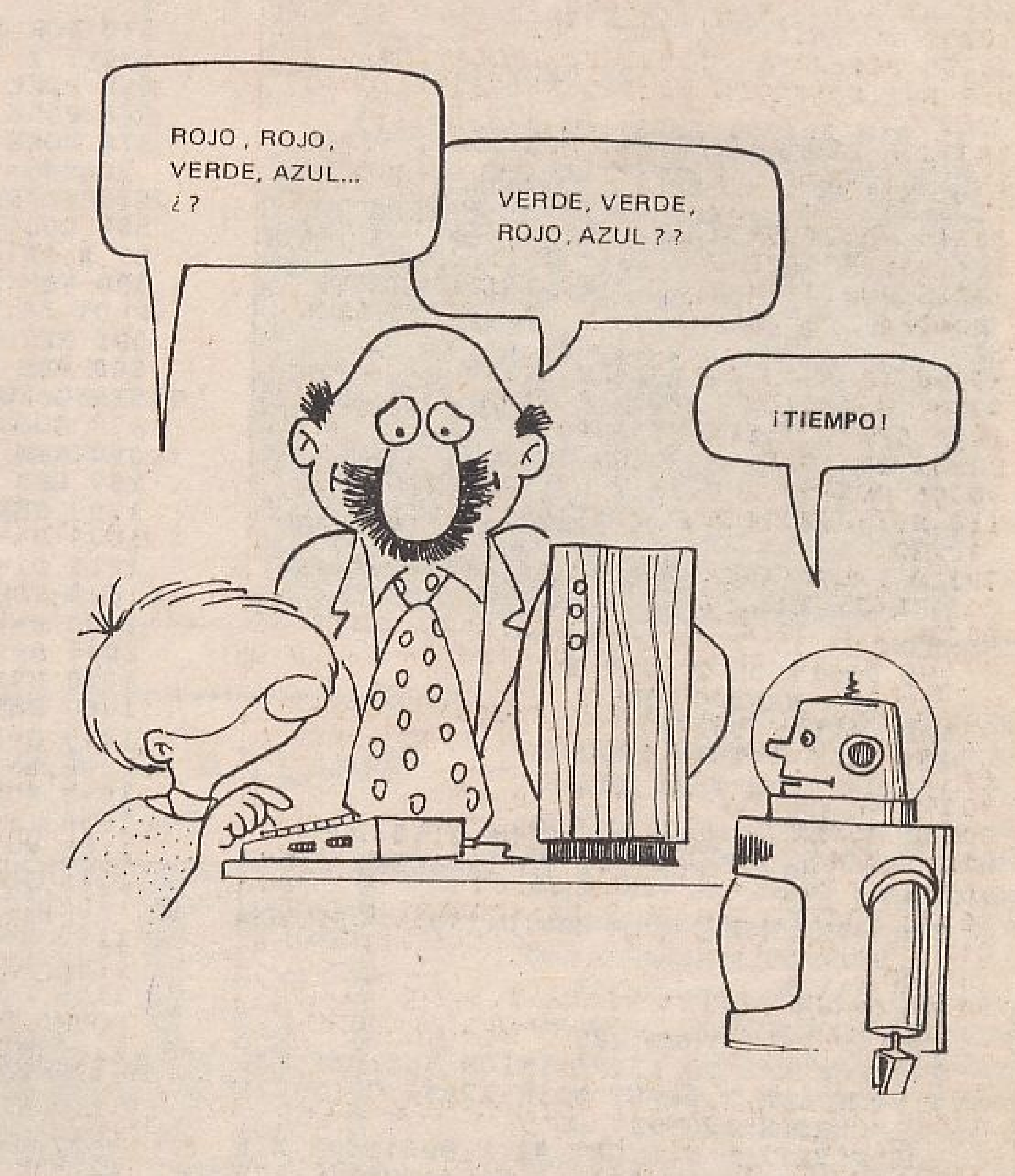
Memorix

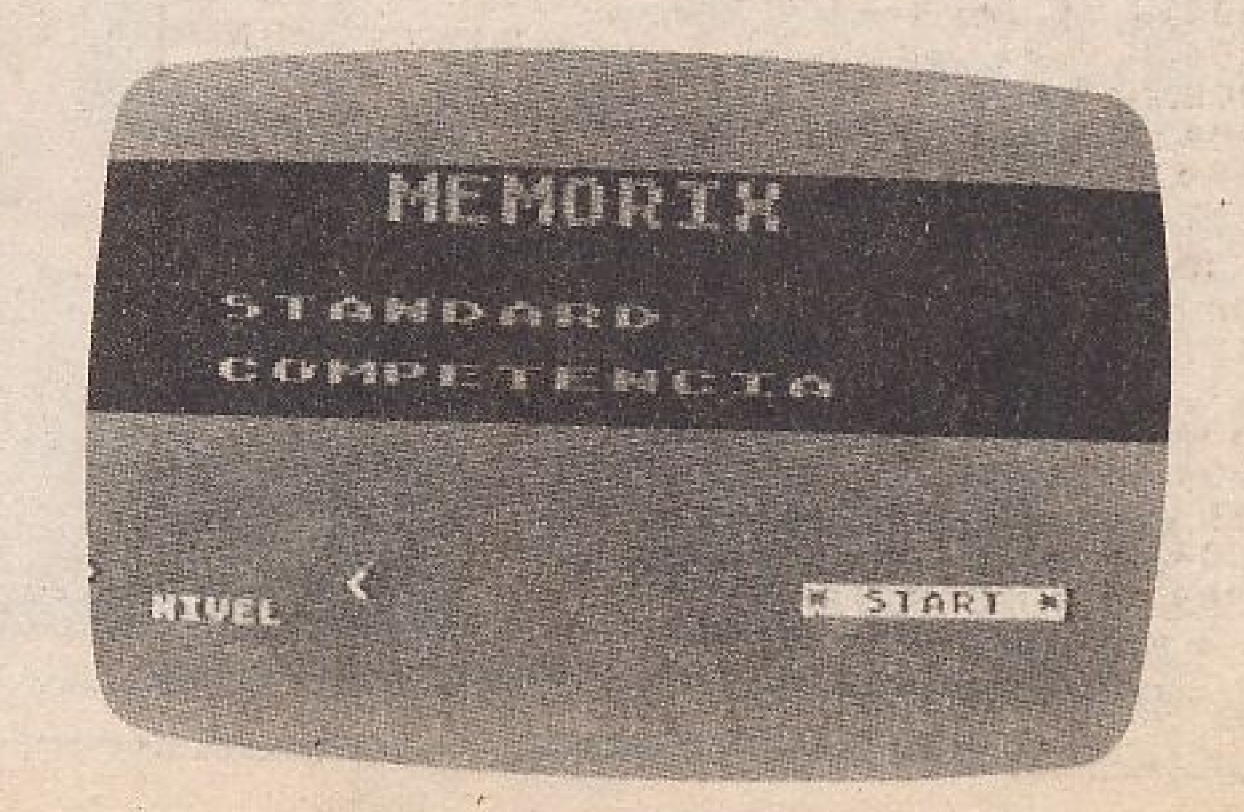
CONFIGURACION REQUERIDA

- *Computador 800 XL, 130 XE ó 600 XL (ver modificaciones a la vuelta)*
- *Casetera o diskettera*

Cada mes publicaremos un juego no bélico que desarrolle habilidades.

Se aceptan contribuciones de nuestros lectores.





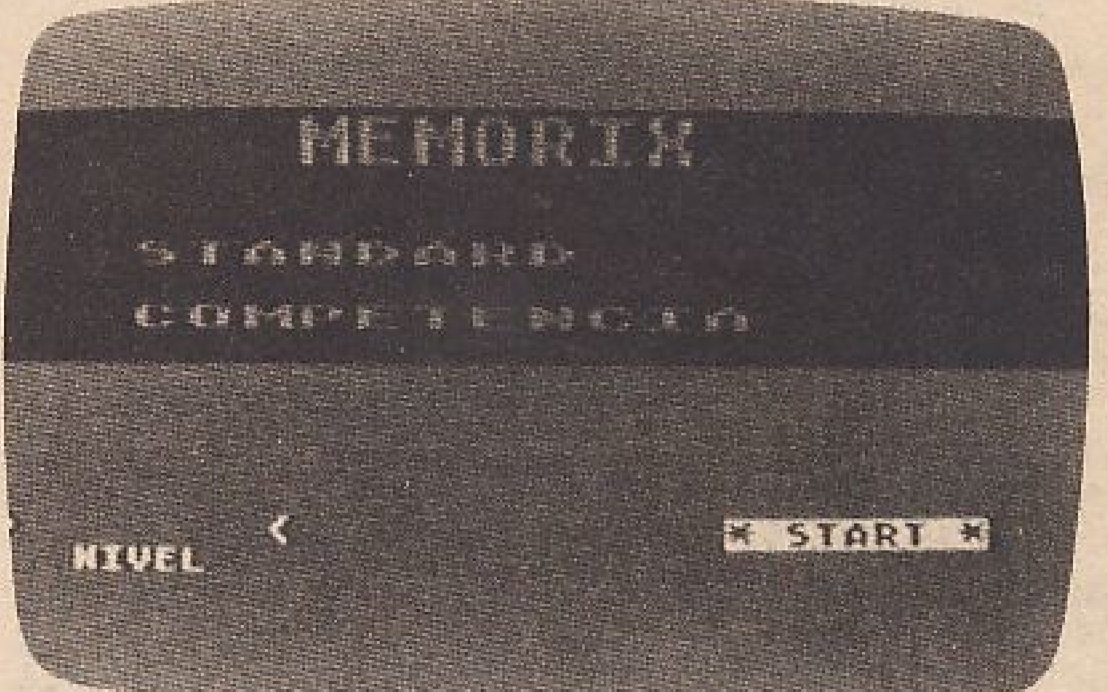
**) Para adaptar el juego al ATARI 600 XL se debe hacer los siguientes cambios:*

- *Eliminar las líneas 512 y 575*
- *Los 10 primeros DATA de línea 520 quedan así: 112, 112, 112, 71, 98, 63, 6, 74, 160, 59 ...*
- *Los 6 últimos DATA de línea 530 quedan así: ... 139, 63, 6, 65, 106, 59*

Eso es todo... ¡Que se diviertan!

```
10 DIM A$(20),B$(9),C$(11),D$(11)
20 D=0:J=-1:TRAP 40
30 CLOSE #1:OPEN #1,12,0,"S:"
35 C$="STANDARD   ":D$="COMPETENCIA"
40 GRAPHICS 0:POKE 752,1:POKE 710,52:P
OKE 708,36
45 FOR Z=0 TO 3:POKE 53248+Z,0:NEXT Z:
SOUND 0,0,0,0
50 DL=PEEK(560)+PEEK(561)*256
60 POKE DL+9,7:POKE DL+10,7:POKE DL+11
,6:POKE DL+12,6:POKE DL+13,6:POKE DL+1
4,6
70 POKE DL+27,112:POKE DL+28,112:POKE
DL+29,112
80 POSITION 6,4:? "MEMORIX":POSITION 3
,5:? C$:POSITION 3,6:? D$
90 POSITION 4,13:? "NIVEL"
100 POSITION 1,12:? ">                <":POSI
TION 27,12:? "* START *"
105 POSITION 2,12:? B$
110 REM MANEJO MENU
130 P=PEEK(53279)
150 IF P=7 THEN 130
160 IF P<>3 THEN 230
170 D=D+1
190 IF D=10 THEN POSITION 2,12:? "
   ":B$="":FOR T=1 TO 150:NEXT T:D=0
:GOTO 130
200 B$(LEN(B$)+1)=CHR$(D+176):POSITION
 2,12:? B$
210 A=1^1
220 GOTO 130
230 IF P<>5 THEN 280
240 FOR X=3 TO 13:POSITION X,5:GET #1,
K:POSITION X,5:PUT #1,K-128*J:C$(X-2,X
-2)=CHR$(K-128*J)
250 POSITION X,6:GET #1,K:POSITION X,6
:PUT #1,K+128*J:D$(X-2,X-2)=CHR$(K+128
*J)
260 NEXT X
270 J=-J
280 IF P=6 AND J=-1 THEN 600
285 IF P=6 AND J=1 THEN 1730
290 GOTO 130
291 STOP
300 REM
310 REM  DISENO DE PLAYER
320 MEMTOP=PEEK(741)+PEEK(742)*256-1
330 PMBASE=INT((MEMTOP-1024)/1024)*102
4
340 ADJTOP=PMBASE+384
350 POKE 742,INT(ADJTOP/256):POKE 742,
ADJTOP-256*PEEK(742)
360 POKE 54279,PMBASE/256
370 POKE 53277,2
380 POKE 559,42
390 P0=PMBASE+512:P1=PMBASE+640:P2=PMB
ASE+768:P3=PMBASE+896
400 FOR Z=37 TO 62:POKE P0+Z,255:POKE
P1+Z,255:NEXT Z
410 FOR Z=67 TO 94:POKE P2+Z,255:POKE
P3+Z,255:NEXT Z
450 FOR X=0 TO 3:POKE 53256+X,3:POKE 7
04+X,0:NEXT X
470 POKE 53248,92:POKE 53249,130:POKE
53250,92:POKE 53251,130
480 RETURN
500 REM TRAZADO PANTALLA
510 GRAPHICS 5:RESTORE
512 LOW=PEEK(88):HIGH=PEEK(89)
520 DATA 112,112,112,71,96,159,6,74,16
0,155,10,10,10,10,10,10,10,10,10,10,10
,10,10,10,10,10,10,10,10,10,10,10,10
530 DATA 10,10,10,10,10,10,10,10,10,10
,10,10,10,10,10,70,137,159,6,65,106,14
9
540 FOR X=1536 TO 1590:READ A:POKE X,A
:NEXT X
550 POKE 560,0:POKE 561,6
560 POKE 656,0:POKE 657,6:? "MEMORIX"
570 POKE 656,1:POKE 657,2:? "tiempo=
jugada="
575 POKE 88,LOW:POKE 89,HIGH
580 COLOR 1:FOR X=20 TO 58:PLOT X,3:PL
OT X,18:PLOT X,34:NEXT X
590 FOR X=3 TO 34:PLOT 20,X:PLOT 39,X:
PLOT 58,X:NEXT X
595 RETURN
600 REM DIRIGE PARTIDA
610 GOSUB 500
620 GOSUB 300
990 REM JUEGO
999 REM MECANISMO DEL JUEGO
1000 REM
1005 F=-1:RESTORE 1230
1010 D1=D*1+8
1020 FOR X=1 TO D1
1030 B=INT(RND(0)*4)+1
1040 A$(X,X)=STR$(B)
1050 NEXT X
1060 REM MUESTRA SECUENCIA MAQUINA
1070 Q=1:X=1
1080 X=X+1
1090 FOR Y=1 TO Q
1095 IF INT(Q/3)<>F THEN READ H
1100 F=INT(Q/3):POKE 540,H
1110 POKE 656,1:POKE 657,10:? " ":POKE
 657,19:POKE 656,1:? Q;" "
1130 B=VAL(A$(Y,Y)):GOSUB 1500
1140 IF PEEK(540) THEN 1140
1150 FOR Z=0 TO 3:POKE 704+Z,0:NEXT Z:
SOUND 0,0,0,0
1160 NEXT Y
1165 REM  REPITE SECUENCIA JUGADOR
1170 FOR Y=1 TO Q
1180 POKE 540,254:POKE 764,255
1190 P=PEEK(764)
1200 IF P<>255 THEN 1300
1205 GOSUB 1720
1210 IF PEEK(540)=0 THEN 1600
1220 GOTO 1190
1230 REM DATOS VELOCIDAD SECUENCIA
1240 DATA 70,70,35,25,18,13,13,13
1300 IF P=14 THEN B=1:GOTO 1350
1310 IF P=15 THEN B=2:GOTO 1350
1320 IF P=6 THEN B=3:GOTO 1350
1330 IF P=7 THEN B=4:GOTO 1350
1340 GOTO 1190
1350 IF J=-1 THEN IF VAL(A$(Y,Y))<>B T
HEN 1550
1360 GOSUB 1500
1380 P=PEEK(53775)
1390 IF P=251 THEN 1380
1400 FOR Z=0 TO 3:POKE 704+Z,0:NEXT Z:
SOUND 0,0,0,0
1405 IF J=1 THEN RETURN
1410 NEXT Y
```

```
1420 Q=Q+1
1430 FOR T=1 TO 300:NEXT T
1440 IF X<=D1 THEN 1080
1450 FOR T=1 TO 6
1460 FOR Z=1 TO 10:POKE 703+B,Z:SOUND
0,100+Z*2,10,8:NEXT Z
1470 NEXT T
1490 GOTO 40
1500 REM CONTROL DESTELLOS Y SONIDOS
1505 POKE 712,6:POKE 708,0
1510 IF B=1 THEN POKE 704,64:SOUND 0,1
21,10,8:RETURN
1520 IF B=2 THEN POKE 705,148:SOUND 0,
96,10,8:RETURN
1530 IF B=3 THEN POKE 706,36:SOUND 0,8
1,10,8:RETURN
1540 POKE 707,192:SOUND 0,60,10,8:RETU
RN
1550 REM RUTINA DE ERROR
1560 GOSUB 1500
1570 SOUND 0,240,6,10
1580 FOR T=1 TO 500:NEXT T
1590 GOSUB 1655:GOTO 40
1600 REM RUTINA TIEMPO=0
1610 FOR Z=1 TO 5
1620 SOUND 0,100,10,14:T=SIN(9):SOUND
0,0,0,0:FOR T=1 TO 80:NEXT T
1630 NEXT Z
1640 SOUND 0,240,6,10:FOR T=1 TO 200:P
OKE 712,T:NEXT T
1650 GOTO 40
1655 REM RUTINA REPETICION
1656 POKE 656,0:POKE 657,22:? "SECUENC
IA CORRECTA"
1658 FOR Z=0 TO 3:POKE 704+Z,0:NEXT Z:
SOUND 0,0,0,0
1660 FOR Y=1 TO Q:B=VAL(A$(Y,Y))
1670 GOSUB 1500
1680 FOR T=1 TO 150:NEXT T
1690 FOR Z=0 TO 3:POKE 704+Z,0:NEXT Z:
SOUND 0,0,0,0
1700 NEXT Y:RETURN
1710 REM SUB CONTROL TIEMPO
1720 S=INT(PEEK(540)/42.5):POKE 656,1:
POKE 657,10:? S:RETURN
1730 REM CONTROL JUEGO COMPETENCIA
1740 A$="":Q=0:PL=1
1750 GOSUB 500:GOSUB 300
1755 POKE 656,0:POKE 657,32:? "PLAYER"
;PL
1760 Q=Q+1:IF PL=1 THEN PL=2:GOTO 1770
1765 PL=1
1770 GOSUB 1180
1780 POKE 656,0:POKE 657,32:? "PLAYER"
;PL
1790 A$(LEN(A$)+1)=STR$(B)
1795 FOR Y=1 TO Q
1797 GOSUB 1180
1800 IF B<>VAL(A$(Y,Y)) THEN 1550
1810 NEXT Y
1820 IF Q>23 THEN 40
1830 GOTO 1760
```



FINALIZO CONCURSO "DE VUELTA A CLASES" CON APUMANQUE Y ATARI



Ejecutivos de COELSA COMPUTACION en la entrega de premios a los colegios ganadores de este año.

Por otro año consecutivo, el Cosmocentro APUMANQUE y ATARI llevaron adelante su concurso "De vuelta a clases", con la activa participación de diversos colegios de Santiago, que competían para obtener uno de los 7 computadores ATARI para los colegios ganadores.

El concurso permite que los alumnos de un colegio que compran útiles escolares y uniformes en Apumanque junten todas las boletas de compras y las entreguen en los Centros de Cómputo ATARI, dispuestos en los pasillos del Cosmocentro, juntando puntajes acumulativos para sus respectivos colegios. De esta forma, la activa participación de los alumnos les permite alcanzar uno de los lugares ganadores.

Los colegios ganadores, entre los más de 60 que participaron fueron:

- **Carmen Teresiano**
- **The English Institute**
- **Redland School**
- **Wenlock**
- **Adventista**
- **Pedro de Valdivia**
- **Compañía de María**

Una vez más ATARI estuvo presente junto a los colegios.



Esta columna permanente está dirigida a todos los lectores para que formen una librería con rutinas, trucos y secretos. Se aceptarán colaboraciones para promover el intercambio de experiencias. Todas las rutinas que sean publicadas recibirán un premio de incentivo consistente en una Tarjeta de Referencia ATARI que contiene todos los datos de interés que un programador debe tener a mano.

```
5 REM DISEÑO DE CARACTERES
10 DATA 128,64,32,16,8,4,2,1
15 GRAPHICS 0:POKE 752,1
20 FOR Y=4 TO 11
25 TRAP 25
30 POSITION 11,20:PRINT "Ingresa valor
   "
35 RESTORE
40 POSITION 25,20:INPUT VALOR:ID=VALOR
45 POSITION 12,Y:PRINT Y-3
50 FOR X=15 TO 22
55 READ BIT
60 IF VALOR>=BIT THEN POSITION X,Y:PRI
NT " ";:VALOR=VALOR-BIT:GOTO 70
65 POSITION X,Y:PRINT " "
70 NEXT X
75 POSITION 25,Y:PRINT ID
80 NEXT Y
85 POSITION 3,20:PRINT "Para una nueva
 figura presione START  o RESET para t
erminar"
90 IF PEEK(53279)<>6 THEN 90
95 PRINT CHR$(125):GOTO 20
```

RUTINA No. 3

CAR.DIS (Diseño de caracteres)

Este utilitario es complementario a la rutina No. 2 (MUNDOATARI No. 0), pero actúa en forma inversa. CAR.DIS permite graficar caracteres a partir de valores entre 0 y 255, para un total de 8 líneas.

CAR.DIS pide ingresar un valor, luego efectúa la transformación a código binario y dibuja en pantalla la línea dada (ver sistema binario en columna ASSEMBLER de este número).

Al terminar la última línea se puede repetir el proceso presionando START o terminar con la tecla BREAK.

Con algunas pequeñas modificaciones, que puede efectuar usted mismo, puede investigar el set de caracteres del computador apuntando a la memoria ROM, a un bloque de localizaciones a partir de la 57344.

RUTINA No. 4

MANEJO DE STRING

Las variables alfanuméricas en ATARI BASIC presentan la particular flexibilidad de poder dimensionarse en el límite máximo de la capacidad de memoria. Esta particularidad **no se encuentra** en otros microcomputadores.

Antes mencionaremos un truco de utilidad en la inicialización de una variable de gran tamaño.

Digite el programa siguiente, ejecute y controle el tiempo que demora en inicializarse el string:

10 DIM ARRE$(20000)
20 FOR J = 1 TO 20000: ARRE$(J) = " ": NEXT J
30 PRINT ARRE$

Ahora digite este nuevo programa y compárelo con el anterior:

10 DIM ARRE$(20000)
20 ARRE$(1) = " ": ARRE$(20000) = " ": ARRE$ (2) = ARRE$
30 PRINT ARRE$

La rutina de este mes presenta en la línea 20 la modificación de las características propias del arreglo de string:

N = variable para número de elementos
E = largo de un elemento

Modifique estas variables a su agrado y ejecute.

```
10 REM ARREGLO DE STRING
20 N=10:E=30:A=N*E:GRAPHICS 0
30 DIM ARRE$(A),ELEM$(E)
100 REM VA A CREAR ARREGLO
110 GOSUB 4000
200 REM BUSCA UN ELEMENTO DEL ARREGLO
210 PRINT "DIGITE numero de elemento q
ue busca ":INPUT K
220 GOSUB 4300
230 PRINT ELEM$
240 END
4000 REM SUBRUTINA CREA ARREGLO STRING
4010 ARRE$(1)=" ":ARRE$(A)=" ":ARRE$(2
)=ARRE$
4030 FOR J=1 TO N
4040 PRINT "DIGITE elemento numero ";J
4050 INPUT ELEM$
4060 L=LEN(ELEM$)
4070 INC=(J-1)*E
4080 ARRE$(1+INC,E+INC)=ELEM$
4090 NEXT J
4100 RETURN
4300 REM SUBRUTINA BUSCA UN ELEMENTO
 DEL ARREGLO
4310 REM CLAVE DE INGRESO =K NUMERO DE
L ELEMENTO
4320 ELEM$=""
4330 INC=(K-1)*E
4340 ELEM$=ARRE$(1+INC,E+INC)
4350 RETURN
```

Primeros pasos en ATARI BASIC

Nuestra columna de rutinas coleccionables se incrementa este mes con una que será disfrutada por los lectores de MUNDO ATARI.

Digite el siguiente listado:

```
10 REM LISTADO #1
15 GRAPHICS 0
20 VELOC=3.48
30 MEDIC=18
40 LAPSO=1.65
100 REM FORMULAS
110 DIM TABLA(MEDIC)
120 FOR J=0 TO MEDIC
130 TABLA(J)=0
140 NEXT J
150 FOR J=0 TO MEDIC
160 DISTAN=VELOC*J*LAPSO
170 TABLA(J)=DISTAN
180 NEXT J
200 ? CHR$(125)
210 POSITION 0,0:? "       TABLA DE VAL
ORES DISTANCIA          ";
220 POSITION 4,2:? "   MEDICION     TIE
MPO    DISTANCIA"
230 FOR J=1 TO MEDIC
240 ? ,J,J*LAPSO,TABLA(J)
250 NEXT J
```

Su ejecución despliega en la pantalla los siguientes valores como datos de salida. La foto ilustra esta situación:

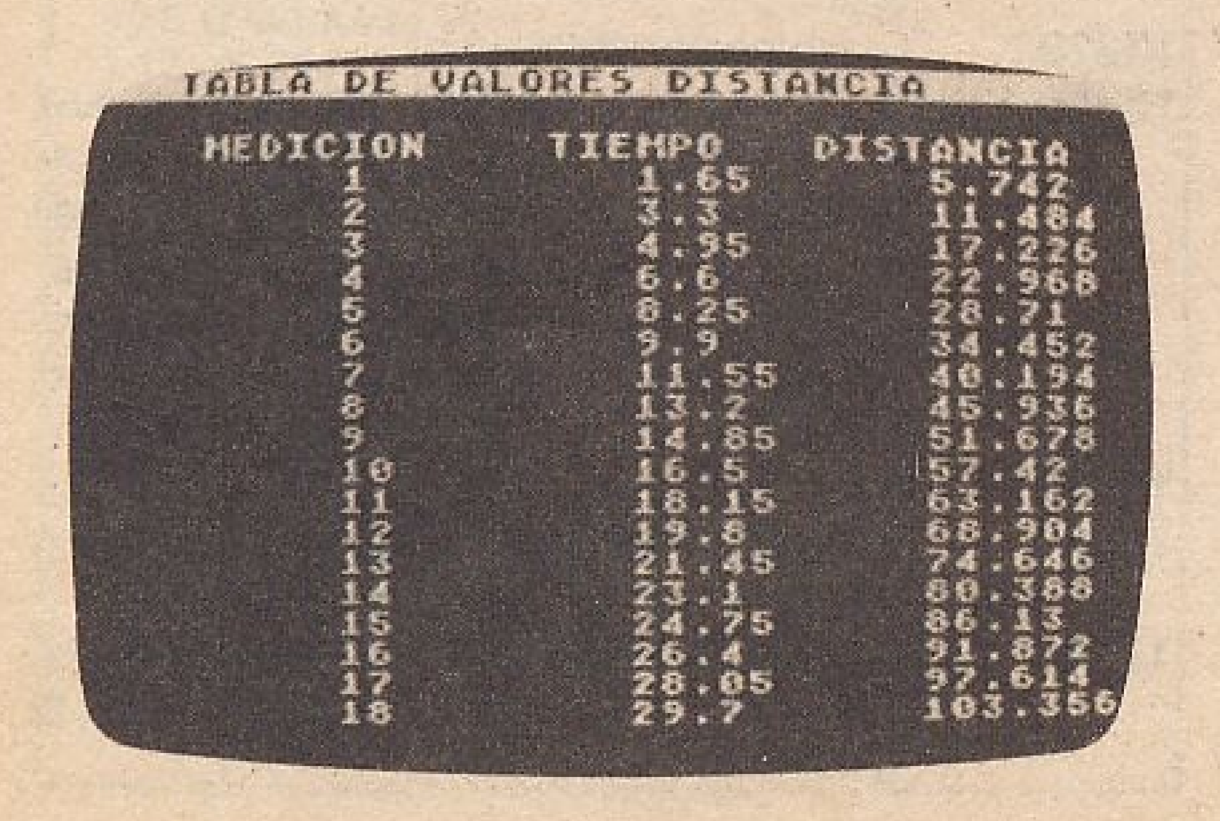
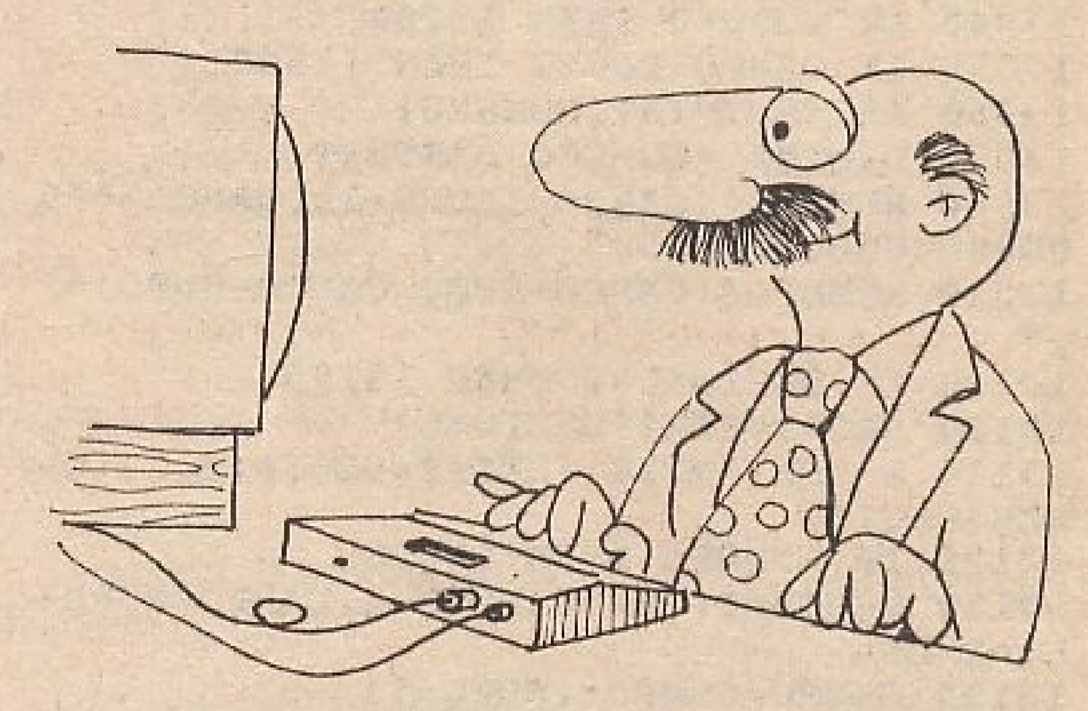


TABLA DE VALORES DISTANCIA

MEDICION	TIEMPO	DISTANCIA
1	1.65	5.742
2	3.3	11.484
3	4.95	17.226
4	6.6	22.968
5	8.25	28.71
6	9.9	34.452
7	11.55	40.194
8	13.2	45.936
9	14.85	51.678
10	16.5	57.42
11	18.15	63.162
12	19.8	68.904
13	21.45	74.646
14	23.1	80.388
15	24.75	86.13
16	26.4	91.872
17	28.05	97.614
18	29.7	103.356



Comentario del listado:

20-40 Inicializa variables de uso en cálculos. Pueden modificarse.

110-140 Dimensiona el arreglo TABLA para el número de mediciones y lo inicializa.

150-180 Ciclo para cálculo de la distancia recorrida por el móvil y almacenamiento de datos en el arreglo.

200-250 Formateo en pantalla de los datos almacenados en el arreglo.

Observemos esta pantalla de datos de las mediciones y evaluemos su presentación:

1. Las columnas se encuentran desplazadas hacia la derecha.
2. Los números de las columnas no guardan la ubicación de acuerdo al dígito. Una mejor ordenación sería la siguiente:

```
   1
  12
   2
1234
```

Estos números se encuentran alineados a la derecha.

3. Los decimales de números no guardan la ubicación con respecto al punto decimal. Ejemplo de una mejor ordenación:

```
  45.670
  12.000
   5.450
1234.000
```

Considerar estos aspectos en la pantalla de sus programas optimizará su presentación.

El listado siguiente corresponde a la rutina de este mes. Aconsejamos tipearla y grabarla en diskette o cassette con instrucción LIST, pues les será útil en alguna oportunidad.

```
13000 REM RUTINA FORMATEO Y PUNTO DECI
MAL DE NUMEROS
13010 REM
13020 TRAP 13150
13030 NUMERO$(1)=" ":NUMERO$(CAMPO)="
":NUMERO$(2)=NUMERO$
13040 IF NUPO=0 THEN 13200
13050 IF CAMPO<DEC+2 THEN 13500
13060 PARTENT=INT(NUMERO)
13070 DECIMAL=NUMERO-PARTENT
13080 NUMERO$(CAMPO-(NUPO-1),CAMPO)="0
0000000000000000"
13090 NUMERO$(CAMPO-NUPO,CAMPO-NUPO)="
,"
13100 IF DECIMAL=0 THEN 13190
13110 DECI$=STR$(DECIMAL)
13120 L=LEN(DECI$):DECI$=DECI$(3,L):L=
LEN(DECI$)
13125 IF L>NUPO THEN L=NUPO
13130 NUMERO$(CAMPO-NUPO+1,CAMPO-NUPO+
L)=DECI$
13190 CAMPO=CAMPO-(NUPO+1)
13200 PARTE$=STR$(PARTENT)
13210 L=LEN(PARTE$)
13220 IF L>CAMPO THEN 13550
13300 CUENTA=1
13310 FOR J=L TO 1 STEP -1
13320 NUMERO$(CAMPO,CAMPO)=PARTE$(J,J)
13325 IF J<2 THEN 13390
13330 K=INT(CUENTA/3)
13340 IF K*3<>CUENTA THEN 13380
13350 CAMPO=CAMPO-1
13360 NUMERO$(CAMPO,CAMPO)="."
13380 CAMPO=CAMPO-1
13390 CUENTA=CUENTA+1:NEXT J
13395 IF S=0 THEN 13450
13400 POSITION X,Y:? NUMERO$
13410 POSITION X,Y:? NUMERO$
13450 IF P=0 THEN RETURN
13460 LINEA$(1)=" ":LINEA$(79)=" ":LIN
EA$(2)=LINEA$
13470 L=LEN(NUMERO$):LINEA$(PL,PL+L-1)
=NUMERO$
13480 LPRINT LINEA$
13490 RETURN
13500 POSITION 0,20:? " ERROR en datos
 ENTRADA a RUTINA";
13510 RETURN
```

Variables numéricas:

NUMERO Ingresa el número a la subrutina

NUPO Ingresa el número de decimales para el formateo del número

CAMPO Ingresa el número de espacios de la pantalla para el formateo de esta columna

X Define la posición en la línea que se edita el numero formateado

Y Define el número de la línea a editar

CUENTA Controla la ubicación para la parte entera del número y la ubicación de los puntos separadores

PARTENT Parte entera del número ingresado a la rutina

DECIMAL Decimal del número, si existe

K Variable de control para ubicar la posición y editar el punto si corresponde

Variables alfanuméricas:

NUMERO$ Contiene el número formateado a editar (salida)

DECI$ Contiene la parte decimal del número

PARTE$ Contiene la parte entera del número

LINEA$ Para formatear el número en una línea de salida a impresor. Puede modificarse a 136 si fuera comprimido.

Comentarios del listado:

13030 Inicializa la variable NUMERO$ para el largo definido en el campo

13040 Direcciona si el número se formatea sin fracciones

13050 Establece condición si se definió incorrectamente los valores de entrada

13060 Asigna a variable PARTENT la parte entera del número

13070 Asigna a variable DECIMAL la parte decimal

13080 Inicializa con 0 para la fracción. Se cubre el caso que los dígitos del decimal sean inferiores a los pedidos en el formateo

13090 Ubica la coma de separación de la parte decimal

13100 Condiciona si el número no presenta parte decimal

13110 Asigna a variable DECI$ la parte decimal

13120 Modifica la variable DECI$ para eliminar los 2 primeros caracteres (0)

13125 Controla un posible error por defecto de variable L

13130 Traslada la parte decimal al número

13140 Modifica el valor de variable CAMPO, para poder ubicar la parte entera en el número

13200 Asigna a variable PARTE$ la parte entera

13220 Establece posible error

13300-13390 Ciclo para trasladar los dígitos desde la variable PARTE$ hacia la variable NUMERO$. Utiliza el STEP–1 para desplazarse de derecha a izquierda. Utiliza la variable CUENTA para identificar la ubicación del punto si corresponde

13410 Edita el valor en la pantalla

13460 Prepara la variable LINEA$ para imprimir la línea

13470 Inserta en la ubicación definida por variable PL el formato del número en la LINEA$

13480 Edita la variable LINEA$ en impresora

Veamos qué sucede con esta subrutina. Digite estas líneas sin borrar de memoria la subrutina de formateo.

```
5 ? CHR$(125)
8 DIM NUMERO$(20),DECI$(18),PARTE$(20)
,LINEA$(80)
10 FOR N=1 TO 6
20 READ NUMERO,CAMPO,X,Y,NUPO,P,S,PL
30 DATA 1234.567,20,5,5,2,1,1,40
31 DATA 12345.67,20,5,6,2,1,1,40
32 DATA 234.567,20,5,7,2,1,1,40
33 DATA 4.567,20,5,8,2,1,1,40
34 DATA 0.567,20,5,9,2,1,1,40
35 DATA 1111111234,20,5,10,2,1,1,40
40 GOSUB 13000
100 NEXT N
1000 END
```

Comentario de líneas adicionales:

5 Limpia la pantalla
8 Dimensiona las variables alfanuméricas que utiliza en subrutina
10 Ciclo para leer valores diferentes de instrucciones DATA
30-35 Líneas con parámetros de entrada a subrutina
40 Direcciona hacia subrutina
1000 Finaliza

Observe la precisión del formateo.

Pero ahora volviendo al punto de partida del listado 1 de la tabla de valores ¿Cómo mejorarla? Bueno, aquí está el listado siguiente:

```
5 DIM NUMERO$(20),DECI$(18),PARTE$(20)
,LINEA$(80)
10 DIM TEMP$(10)
15 GRAPHICS 0
20 VELOC=3.48
30 MEDIC=18
40 LAPSO=1.65
100 REM FORMULAS
110 DIM TABLA(MEDIC)
120 FOR J=0 TO MEDIC
130 TABLA(J)=0
140 NEXT J
150 FOR J=0 TO MEDIC
160 DISTAN=VELOC*J*LAPSO
170 TABLA(J)=DISTAN
180 NEXT J
200 ? CHR$(125)
210 POSITION 0,0:? "        TABLA DE VAL
ORES DISTANCIA          ";
220 POSITION 0,2:? "  MEDICION    TIE
MPO   DISTANCIA"
300 FOR ELE=1 TO MEDIC
310 TEMP$=STR$(ELE)
320 L=LEN(TEMP$)
330 POSITION 11-L,ELE+2:? TEMP$;
400 CAMPO=6
410 NUPO=2
420 X=15
430 Y=ELE+2
440 P=0
450 S=1
460 PL=0
470 NUMERO=ELE*LAPSO
500 GOSUB 13000
510 CAMPO=12
520 NUPO=3
530 X=21
540 Y=ELE+2
550 P=0
560 S=1
570 PL=0
580 NUMERO=TABLA(ELE)
600 GOSUB 13000
650 NEXT ELE
1000 END
```

Digítelo.

Ingrese con instrucción ENTER la subrutina FORMATEO:

ENTER "C" **(para casetera)**
ENTER "D: FORMATEO.NUM" **(para diskettera)**

Digite la instrucción LIST y su programa se combinará con el anterior.

Ejecute el programa y compare con la presentación inicial de la pantalla. La foto ilustra estos aspectos:

TABLA DE VALORES DISTANCIA

MEDICION	TIEMPO	DISTANCIA
1	1,65	5,742
2	3,30	11,484
3	4,95	17,226
4	6,60	22,968
5	8,25	28,710
6	9,90	34,452
7	11,55	40,194
8	13,20	45,936
9	14,85	51,678
10	16,50	57,420
11	18,15	63,162
12	19,80	68,904
13	21,45	74,646
14	23,10	80,388
15	24,75	86,130
16	26,40	91,872
17	28,05	97,614
18	29,70	103,356

Si usted tiene una impresora como parte de su configuración podrá experimentar con esta subrutina para salidas hacia documentos escritos.

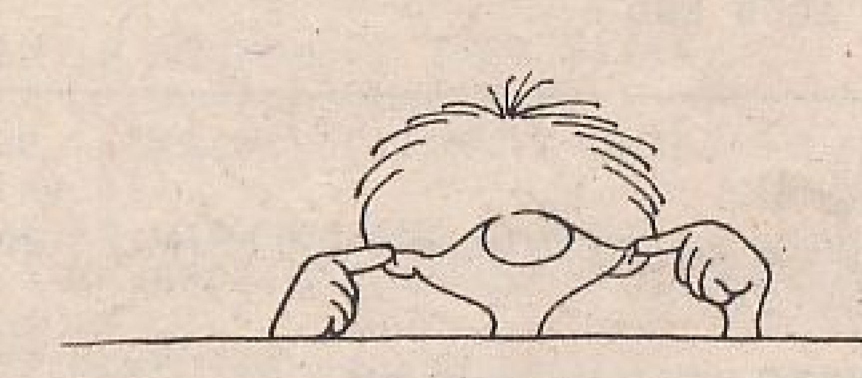
ACTIVIDADES DE SUPERACION

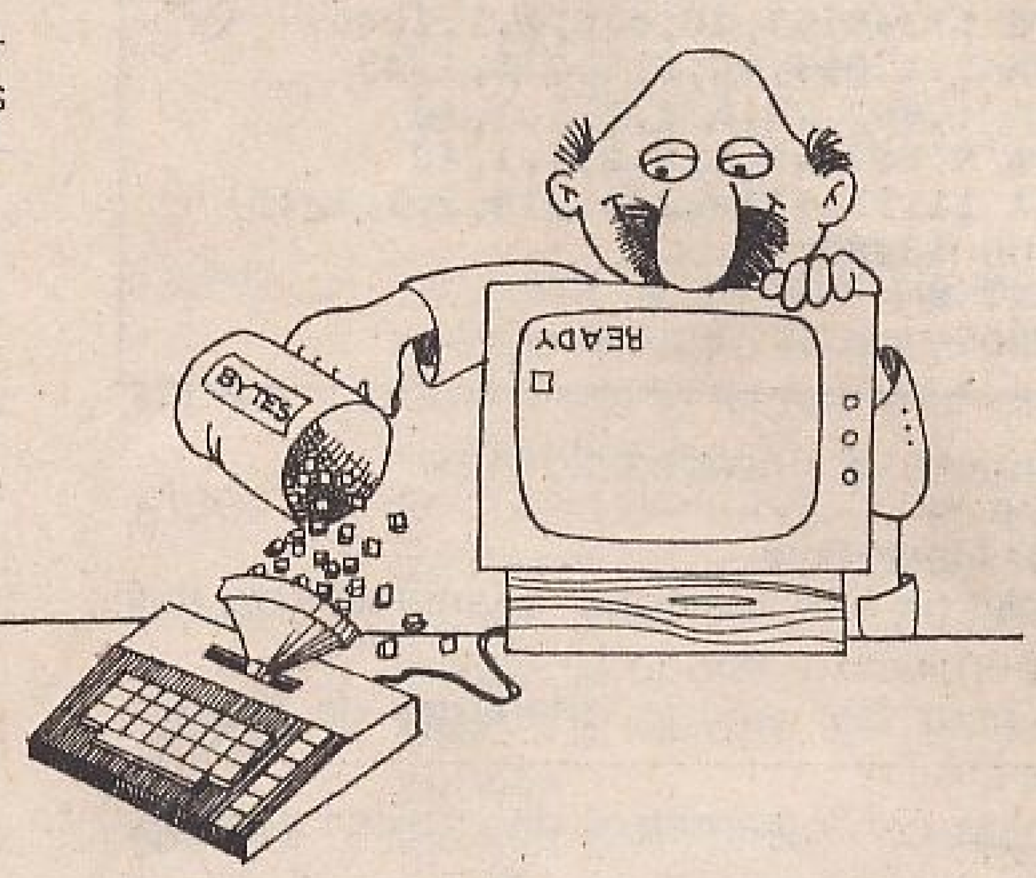
Esta rutina no contempla aproximaciones, pueden ser implementadas llamando a otra subrutina y estableciendo variables de señal para direccionar la posibilidad.

Trabaje en ella y envíela como cooperación a nuestra columna. Hay premios para usted. ●

De byte en byte

Hoy continuamos con el análisis de 3 localizaciones que controlan la pantalla y que son las predilectas del Profesor Chip.





6. MODO GRAFICO
Localización 87

Esta localización mantiene el registro correspondiente al modo gráfico en uso.

Digite las siguientes líneas:

```
10 GRAPHICS 21: REM Modo gráfico 5 sin ventana de texto
20 DL=PEEK(560)+256 * PEEK(561): REM Asigna a variable DL localización inicial de Lista de Despliegue
30 POKE DL + 6, 2: REM Modifica una línea a Modo cero
50 PRINT "MUNDOATARI"
```

Nota: La columna TECNICAS de MUNDOATARI desarrolla el tema Lista de Despliegue.

ACTIVIDAD:

Ejecute el listado y observe el comportamiento. ¿En qué modo gráfico edita el texto MUNDOATARI? ¿Qué pasó con el Modo 5?

Para comprobar la utilidad de la localización 87 agregue las siguientes líneas:

```
35 POKE 87, 0: REM Al modificar esta localización con el valor cero se engaña al Sistema Operativo
40 POSITION 20, 0
```

ACTIVIDAD:

Ejecute el listado nuevamente con las nuevas líneas. ¿Qué diferencias observa con la ejecución anterior?

7. PRIMERA LOCALIZACION DE PANTALLA
Localizaciones 88 y 89

Este puntero mantiene el registro que corresponde al primer carácter editado en la pantalla. (ver página 33 de MUNDOATARI No. 0).

Digite el siguiente listado:

```
10 GRAPHICS 0
20 LOCI = PEEK(88) + 256 * PEEK(89): REM Asigna a variable LOCI el puntero inicial de pantalla.
30 FOR J=0 TO 25
40 POKE LOCI + J, J + 33
50 NEXT J
```

Las líneas 30, 40 y 50 ejecutan operaciones directas a la máquina, modificando las localizaciones que corresponden a la pantalla.

El valor agregado de 33 es para proporcionar la clave de transformación de modo de editar los caracteres deseados (en este caso las letras mayúsculas).

Ejecute el programa y observe la pantalla. ¿Qué pasa con la letra C?

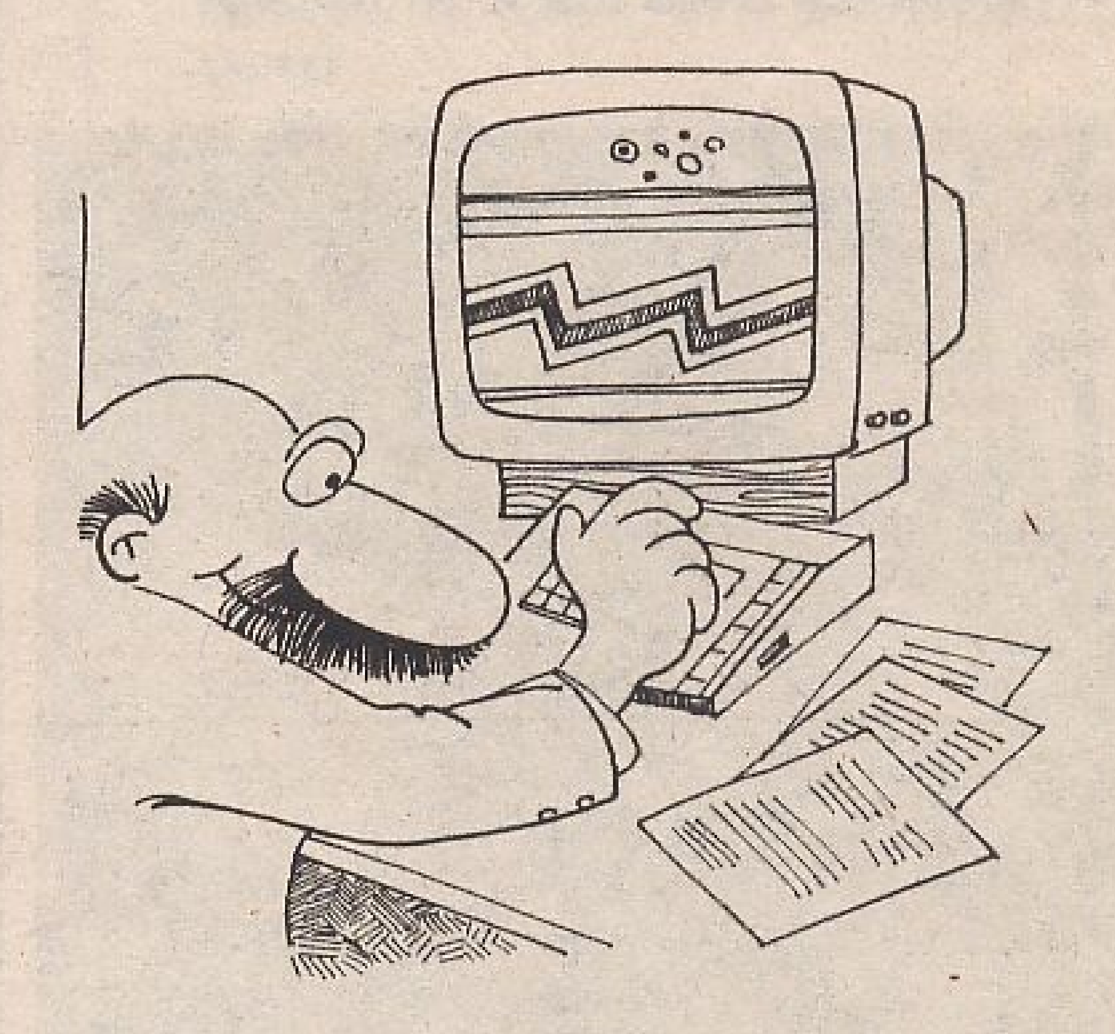
Agregue la siguiente línea:

```
15 POSITION 0, 5: PRINT " "
```

¿Cuál es la diferencia en el comportamiento? ¿Por qué?

En el próximo número continúa el análisis de las localizaciones que se relacionan con la pantalla.

Lista de Despliegue



Hoy continuamos la columna **Lista de Despliegue** analizando en particular los tipos de instrucciones que forman la Lista.

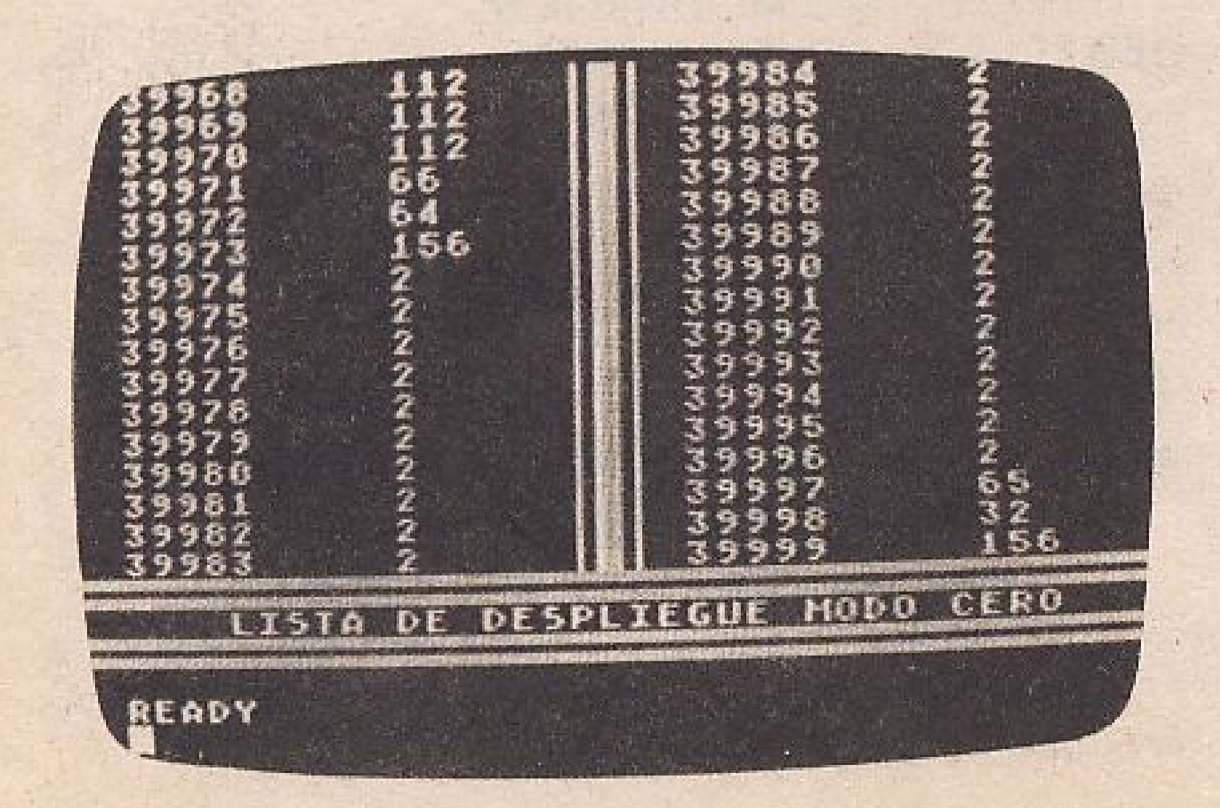
En el número anterior se analizó en extenso:

a) Definición de Lista de Despliegue.
b) Lista de Despliegue para el Modo 0.
c) Lista para otros modos gráficos.
d) Puntero que indica localización inicial para la Lista de Despliegue.

Tipee el siguiente listado:

```
10 REM
20 GRAPHICS 0
30 INICIAL=PEEK(560)+256*PEEK(561)
40 FOR J=INICIAL TO INICIAL+15
50 ? J,PEEK(J)
60 NEXT J
65 CUENTA=0
70 FOR J=INICIAL+16 TO INICIAL+31
80 POSITION 22,CUENTA:? J,PEEK(J)
85 CUENTA=CUENTA+1
90 NEXT J
100 FOR J=0 TO 15:POSITION 18,J:? "|█|
";:NEXT J
110 POSITION 0,16:? "════════════════════
═══════════════════";
120 ? "    LISTA DE DESPLIEGUE MODO CE
RO  "
130 POSITION 0,18:? "════════════════════
═══════════════════";
```

Al ejecutar este listado aparece en pantalla:



Observe el formato de presentación para que las instrucciones se analicen todas en una sola pantalla.

Comentario del listado:

30 Asigna a variable INICIAL la localización de memoria inicial para la Lista en Modo 0.
40-60 Despliega las primeras 16 instrucciones en dos columnas (localización de memoria y valor)
65-90 Despliega las 16 instrucciones finales en la mitad derecha de la pantalla.
100 Dibuja para cada línea un separador.
110-130 Mensajes para identificar el despliegue.

En esta lista de despliegue aparecen los tipos de instrucciones:

a) Líneas Scan en blanco.
b) Instrucción LMS.
c) Instrucciones ANTIC.
d) Instrucción JUMP (salto de retorno)

A. LINEAS SCAN EN BLANCO

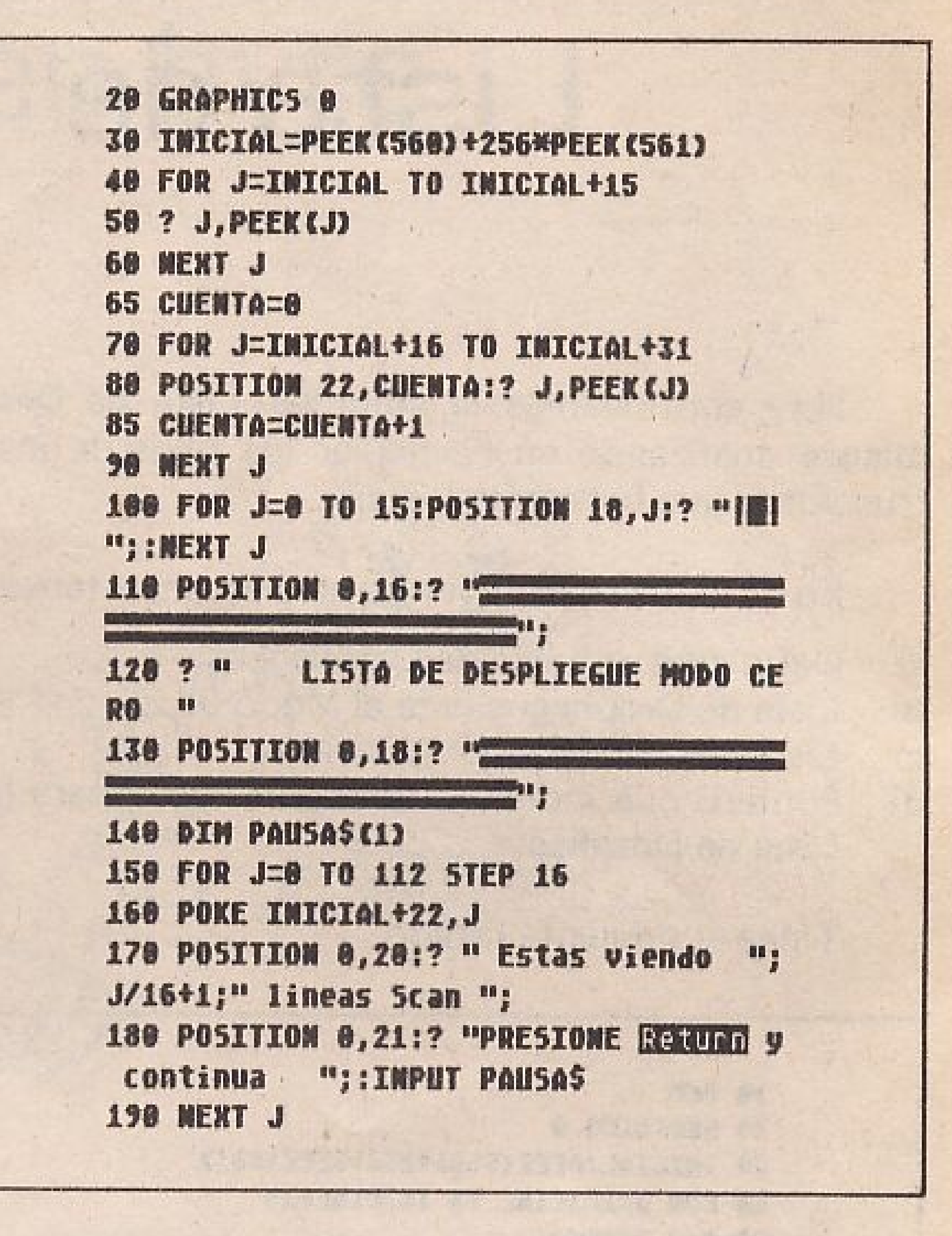
Son las instrucciones para las cuales el ANTIC no requiere de memoria para desplegarlas.

Existen diferentes valores para desplegar de 1 a 8 líneas scan en blanco.

En la lista de pantalla aparecen 3, y corresponden a los valores 112.

Se utilizan para compensar el despliegue total de la pantalla. Sirven como separadores.

Digite las siguientes líneas:



```
20 GRAPHICS 0
30 INICIAL=PEEK(560)+256*PEEK(561)
40 FOR J=INICIAL TO INICIAL+15
50 ? J,PEEK(J)
60 NEXT J
65 CUENTA=0
70 FOR J=INICIAL+16 TO INICIAL+31
80 POSITION 22,CUENTA:? J,PEEK(J)
85 CUENTA=CUENTA+1
90 NEXT J
100 FOR J=0 TO 15:POSITION 18,J:? "|█|
";:NEXT J
110 POSITION 0,16:? "══════════════════
═══════════════════";
120 ? "    LISTA DE DESPLIEGUE MODO CE
RO  "
130 POSITION 0,18:? "══════════════════
═══════════════════";
140 DIM PAUSA$(1)
150 FOR J=0 TO 112 STEP 16
160 POKE INICIAL+22,J
170 POSITION 0,20:? " Estas viendo  ";
J/16+1;" lineas Scan ";
180 POSITION 0,21:? "PRESIONE Return y
 continua   ";:INPUT PAUSA$
190 NEXT J
```

Este listado es el mismo Listado 1, al que se le agregan las líneas 140-190.

Ejecútelo y observará el efecto de aplicar las diferentes líneas scan.

Puede observar con más intensidad el efecto cambiando el color del fondo por blanco mediante la instrucción:

145 POKE 712, 15

Comentario de líneas adicionales:

140 Dimensiona variable alfanumérica PAUSA$ para producir una pausa entre cada número de líneas scan.

150 Ciclo para los valores posibles aceptados por ANTIC para las líneas scan.

160 Modificando el contenido de la localización de memoria para línea 18 que corresponde a INICIAL + 22.

170-190 Mensajes para identificar el número de líneas scan e ingresar la pausa.

ACTIVIDADES COMPLEMENTARIAS

1. Modifique la línea 160 en el incremento a variable inicial. Determine cuáles son los límites aceptables.
2. Complete la siguiente tabla con los valores de instrucción de línea blanca scan y número de líneas desplegadas:

No. líneas	valor asociado
1	
2	
3	
4	
5	
6	
7	
8	

B. INSTRUCCION LMS

Es una instrucción que se localiza en 3 bytes consecutivos de memoria.

En la lista en pantalla corresponde a:

66 en localización 39971
64 en localización 39972
156 en localización 39973

Estos valores son diferentes para el 600 XL y corresponden a:

66 en localización 15395
64 en localización 15396
60 en localización 15397

El significado y funcionalidad en ambos equipos es idéntica sólo varía la ubicación en memoria.

Como dato informativo puede transformar su ATARI 800 en 600 ingresando la línea:

5 POKE 106, 64

ACTIVIDAD COMPLEMENTARIA

Investigue y justifique la transformación. La localización 106 es muy útil para las ocasiones que necesite reservar memoria.

PRIMER BYTE

Corresponde a una suma de un valor fijo más el valor ANTIC para el modo definido vía BASIC con la instrucción GRAPHICS.

En este caso particular el modo definido es el cero. La instrucción ANTIC asociada a este modo el 2.

¿Cuál es el valor fijo entonces?

66 – 2 = ???

64 es el valor fjjo, que sumado al valor de la instrucción ANTIC determina el primer byte de una instrucción LMS.

¿Para qué sirve una instrucción LMS?

Para identificar el block o área de memoria en que se encuentran los datos que se proyectarán hacia la pantalla:

BYTE 2 + 256 * BYTE 3 = dirección inicial de memoria para la pantalla

Para el ATARI 800 corresponde a:

64 + 256 * 156 = 40.000

Para el ATARI 600 corresponde a:

64 + 256 * 60 = 15.424

Digite el Listado 3:

```
15 PRINT CHR$(125)
20 INICIAL=PEEK(560)+256*PEEK(561)
30 REM INICIAL+3=BYTE 1 DE LMS
40 REM INICIAL+4=BYTE 2 DE LMS
50 REM INICIAL+5=BYTE 3 DE LMS
60 POSITION 0,2:? "BYTE 1 =";PEEK(INIC
IAL+3)
70 POSITION 0,3:? "BYTE 2 =";PEEK(INIC
IAL+4)
80 POSITION 0,4:? "BYTE 3 =";PEEK(INIC
IAL+5)
90 POSITION 0,6:? "  BYTE 2 + 256*BYTE
 3     =LOC. PANTALLA"
100 POSITION 4,7:? PEEK(INICIAL+4);"
",;PEEK(INICIAL+5);," = ";PEEK(INICIAL
+4)+256*PEEK(INICIAL+5)
```

Al ejecutar el listado comprobará lo afirmado con respecto a los valores de pantalla para su ATARI

Comentario del listado:

20 INICIAL = variable para inicio de Lista.
30-50 Ubicación de bytes de LMS en la Lista.
50-80 Edita los valores en sus localizaciones.
90-100 Edita el valor inicial para su ATARI.

Pero ahora ¿Cómo interactuar con estos valores para que este estudio tenga una aplicación práctica?

Ahora tiene un anticipo:

Digite en modo directo:

POKE 40000, 33

En la posición 0, 0 aparece en pantalla una letra A.

Digite a continuación:

POKE 40000, 0

Y lo que había en pantalla ya no está.

ACTIVIDADES COMPLEMENTARIAS

1. ¿Qué relación tienen estos valores que se ingresan en la memoria de pantalla con los caracteres editados?
2. Edite en la primera línea de pantalla, pokeando directamente a la memoria, un mensaje como:

HOLA MUNDO ATARI

El próximo mes continuaremos con las instrucciones ANTIC y JUMP y finalizaremos esta técnica con algunos ejemplos de modificaciones a la Lista de Despliegue. ●

Directo al 6502

Esta columna de continuidad desarrollará los conceptos básicos de programación en Assembler.

SISTEMA NUMERICO BINARIO

La Unidad Central de Proceso interpreta sólo dos estados posibles:

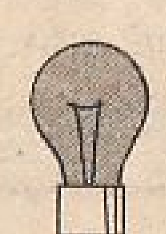
- con paso de corriente
- sin paso de corriente

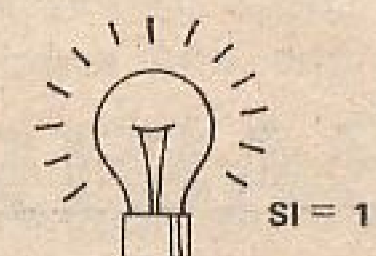
Para una mejor comprensión representaremos esto con una ampolleta en que el paso de corriente genera la luminosidad.

El estado de **no paso** se representa con la ampolleta apagada.

Asociamos a estos dos estados los dígitos:

- el 0 corresponde al **no paso** de corriente
- el 1 indica **paso** de corriente.





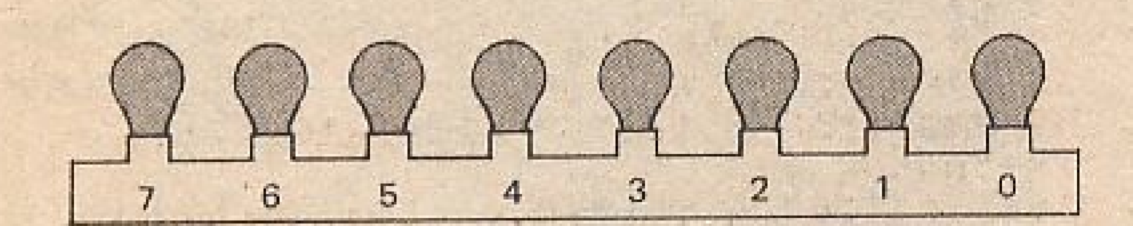
El computador interpreta un número definido de señales. Este número corresponde a ocho.

Cada elemento básico (señal) se denomina **bit** (trozo). El conjunto de ocho bits constituye un **byte**.

Representemos al byte por un conjunto de ocho ampolletas.

La primera posibilidad es que se encuentren todas apagadas, como sucede en la figura.

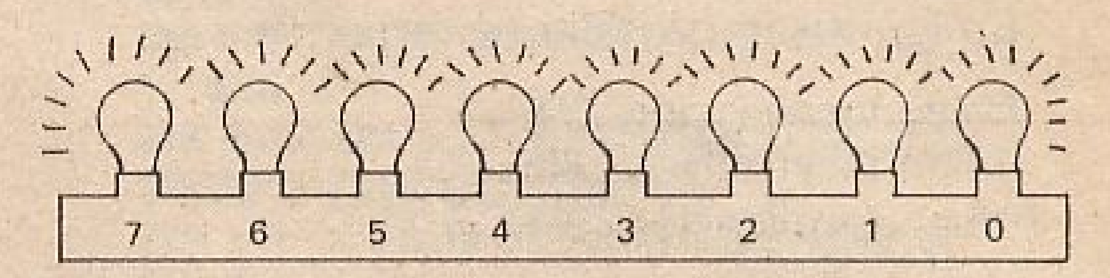
Para designar las ampolletas (bits) las numeramos de derecha a izquierda (0 al 7).



Revise el concepto **base de un sistema numérico** en el número 0 de MUNDOATARI, página 30.

En el **Sistema Binario** existen dos dígitos: **0** y **1**. Por lo tanto la base de este sistema es dos (2).

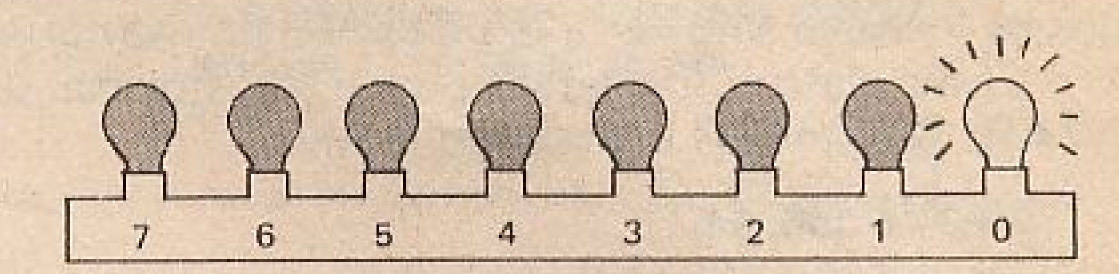
Observe la figura siguiente:



La CPU las interpreta como todas encendidas. ¿Pero cuál es la equivalencia en nuestro sistema numérico decimal?

Cada bit equivale a un determinado valor decimal.

Ejemplo:



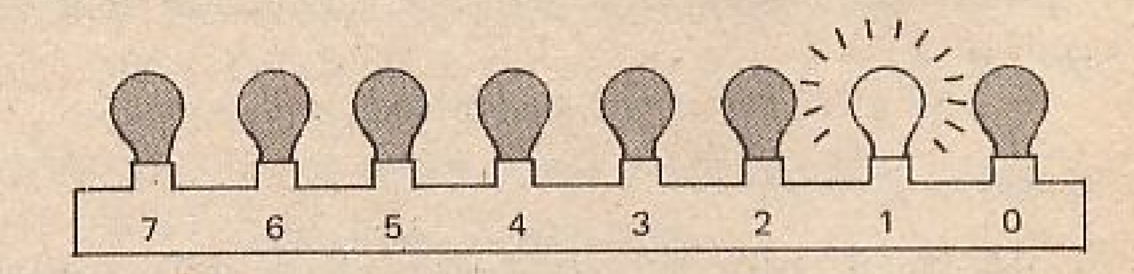
Revise los conceptos de valor asociados a la posición (número 0 de MUNDOATARI).

El valor asociado es la **base** del sistema **elevado** al número de posición:

$2^0 = 1$

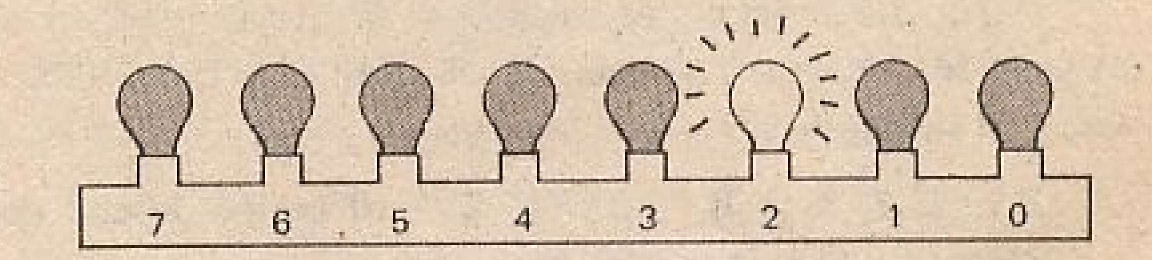
La equivalencia a decimal de este ejemplo corresponde al decimal 1.

Para el caso de la segunda ampolleta encendida:



el valor asociado es: $2^1 = 2$

Para la tercera ampolleta encendida:



el valor asociado es: $2^2 = 4$.

El tercer bit (posición 2) corresponde al número 4 de nuestro sistema decimal.

Desarrolle a continuación:

- ¿A qué valor decimal se asocia el cuarto bit? 2^3 = ...
- ¿A qué valor decimal se asocia el quinto bit? 2^4 = ...
- ¿A qué valor decimal se asocia el sexto bit? 2^5 = ...
- ¿A qué valor decimal se asocia el séptimo bit? 2^6 = ...
- ¿A qué valor decimal se asocia el octavo bit? 2^7 = ...

Tenemos en resumen:

bit número							
7	6	5	4	3	2	1	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
valor decimal							

Para conocer el valor asociado a todas las ampolletas encendidas se suman los valores individuales de las posiciones.

128 + 64 +32 + 16 + 8 + 4 + 2 + 1

El valor es 255, que corresponde al máximo valor decimal que se puede interpretar con los ocho bits.

Aplique los conceptos aprendidos para la resolución de los siguientes ejercicios donde representamos a los binarios por una serie de 8 números (0 y 1):

1. ¿Cuál es el decimal asociado a los binarios siguientes?:

0 0 0 0 0 0 0 0 : ....
0 0 0 0 0 0 0 1 : ....
0 0 0 0 0 0 1 0 : ....
0 0 0 0 0 0 1 1 : ....
0 0 0 0 0 1 0 0 : ....
0 0 0 0 0 1 0 1 : ....
0 0 0 0 0 1 1 0 : ....
0 0 0 0 0 1 1 1 : ....
0 0 0 0 1 0 0 0 : ....
0 0 0 0 1 0 0 1 : ....
0 0 0 0 1 0 1 0 : ....
0 0 0 0 1 0 1 1 : ....
0 0 0 0 1 1 0 0 : ....
0 0 0 0 1 1 0 1 : ....
0 0 0 0 1 1 1 0 : ....
0 0 0 0 1 1 1 1 : ....
1 0 0 0 1 0 0 0 : ....
1 1 1 1 0 0 0 0 : ....
1 0 0 1 1 0 0 1 : ....
1 1 0 0 1 1 0 0 : ....

Para comprobar los resultados digite el siguiente programa:

```
10 REM CONVERSION DE BINARIO A DECIMAL
20 DIM A$(8),PAUSA$(1)
30 GRAPHICS 0
40 POSITION 0,2:? " DIGITE EL NUMERO B
INARIO"
50 INPUT A$
60 L=LEN(A$)
70 FOR J=1 TO L
80 NUM=ASC(A$(J,J))
90 IF NUM<48 OR NUM>49 THEN 130
100 NEXT J
110 GOTO 150
130 POSITION 0,18:? " ERROR en DIGITAC
ION del binario";
135 POSITION 0,20:? " PRESIONE Return
y continua"
140 INPUT PAUSA$
145 GOTO 30
150 DECIMAL=0:INDICE=0:FOR J=L TO 1 ST
EP -1
160 DIGITO=VAL(A$(J,J))
170 DECIMAL=DECIMAL+(DIGITO*2^INDICE)
180 INDICE=INDICE+1
190 NEXT J
200 POSITION 0,16:? " El decimal que c
orresponde al binario";
210 POSITION 0,17:? A$;"  es : ";DECIM
AL
250 END
```

2. Escriba el binario correspondiente a los siguientes decimales:

55, 40, 33, 128, 151, 88, 31, 77, 127
163, 200, 215, 96, 244, 202, 17, 255

Para comprobar digite el siguiente listado:

```
10 REM TRANSFORMA DECIMAL EN BINARIO
20 DIM PAUSA$(1),BINARIO$(8)
30 GRAPHICS 0
40 POSITION 0,2:? "DIGITE numero DECIM
AL"
45 TRAP 30
50 INPUT DECIMAL
60 IF DECIMAL<0 OR DECIMAL>255 THEN 30
70 BINARIO$="00000000"
80 DIGITO=1
90 TR=DECIMAL
100 DIGITO=7:FOR J=1 TO 7
110 POT=2^DIGITO
120 IF TR<POT THEN 150
130 TR=TR-POT
140 BINARIO$(J,J)="1"
150 DIGITO=DIGITO-1
160 NEXT J
170 IF TR=0 THEN 200
180 BINARIO$(8,8)="1"
200 POSITION 0,16:? " El BINARIO que c
orresponde al"
210 POSITION 0,18:? " decimal ";DECIMA
L;" es : ";BINARIO$
```

En el próximo número finalizaremos los Sistemas Numéricos con el Sistema Hexadecimal.

USR

Hoy iniciamos esta columna como complemento a DIRECTO AL 6502. Está dirigida a los lectores para que formen una colección de rutinas utilitarias en lenguaje de máquina.

La función USR ejecuta los siguientes pasos:

- desconecta la secuencia del intérprete BASIC,
- realiza un salto hacia un área específica de la memoria, donde se encuentra un corto trabajo en lenguaje de máquina,
- ejecuta dicha rutina en máquina,
- finaliza con un código de retorno hacia la interrupción del BASIC.

El formato de la función USR es:

USR (parám. 1, parám. 2, parám. 3, ...)
(soporta hasta 27 parámetros)

El primer valor que pasa al STACK –área de la memoria que sirve como archivo temporal de información– es la dirección del retorno hacia el BASIC.

El formato: **X = USR (1536, 40000, N)** indica:

a) El primer parámetro señala que la rutina a ejecutar se encuentra codificada a partir de la localización 1536 (página seis).
b) El segundo parámetro corresponde al número 40000
c) El tercer parámetro corresponde al valor de la variable N (en este caso valdrá 512)

La representación del STACK cuando se ejecuta la función será:

Valores en STACK

No. argumentos	menor	2	Orden de salida ↓
Argumento No. 2 (40000)	mayor menor	156 64	
Argumento No. 3 (N=512)	mayor menor	2 0	
Dirección de retorno al BASIC (2 bytes)		. . .	

Los valores del STACK se recuperan desde arriba hacia abajo. El primero es el valor 2, que corresponde al número de argumentos del ejemplo.



Luego viene el valor 40000, descompuesto en byte menor (156) y byte mayor (64). Finalmente el valor de N (512), tambien descompuesto en menor (0) y mayor (2).

El último par de bytes es el puntero que mantiene la localización de memoria para el retorno al programa BASIC.

Para aplicar los conceptos digite y ejecute el siguiente programa:

```
5 REM LISTADO FUNCION USR
10 DIM AZAR$(12)
20 FOR J=1 TO 12
30 READ N
40 AZAR$(J)=CHR$(N)
50 NEXT J
60 LOC=ADR(AZAR$)
70 NUM=USR(LOC)
80 PRINT NUM
90 GOTO 60
100 DATA 104,173,10,210,133,212,173,10
,210,133,213,96
```

Observe los resultados. El programa edita en forma instantánea un número random (al azar) entre 0 y 65535. Nuestra interrogante para nuestros lectores es: ¿Cómo se traspasa el valor obtenido en la variable NUM hacia el BASIC?

Un comentario final es que los conceptos de sistemas numéricos que se tratan en la columna DIRECTO AL 6502 permitirán interpretar los códigos de máquina y darles un sentido en relación a las operaciones de la CPU. Es importante, amigo lector, que se interese por esa información gradual que con el tiempo tendrá aplicación directa con los ejemplos que trataremos en estas páginas. ●

EDUCANDO CON ATARI

PEQUES: **Rapidez de reflejos**

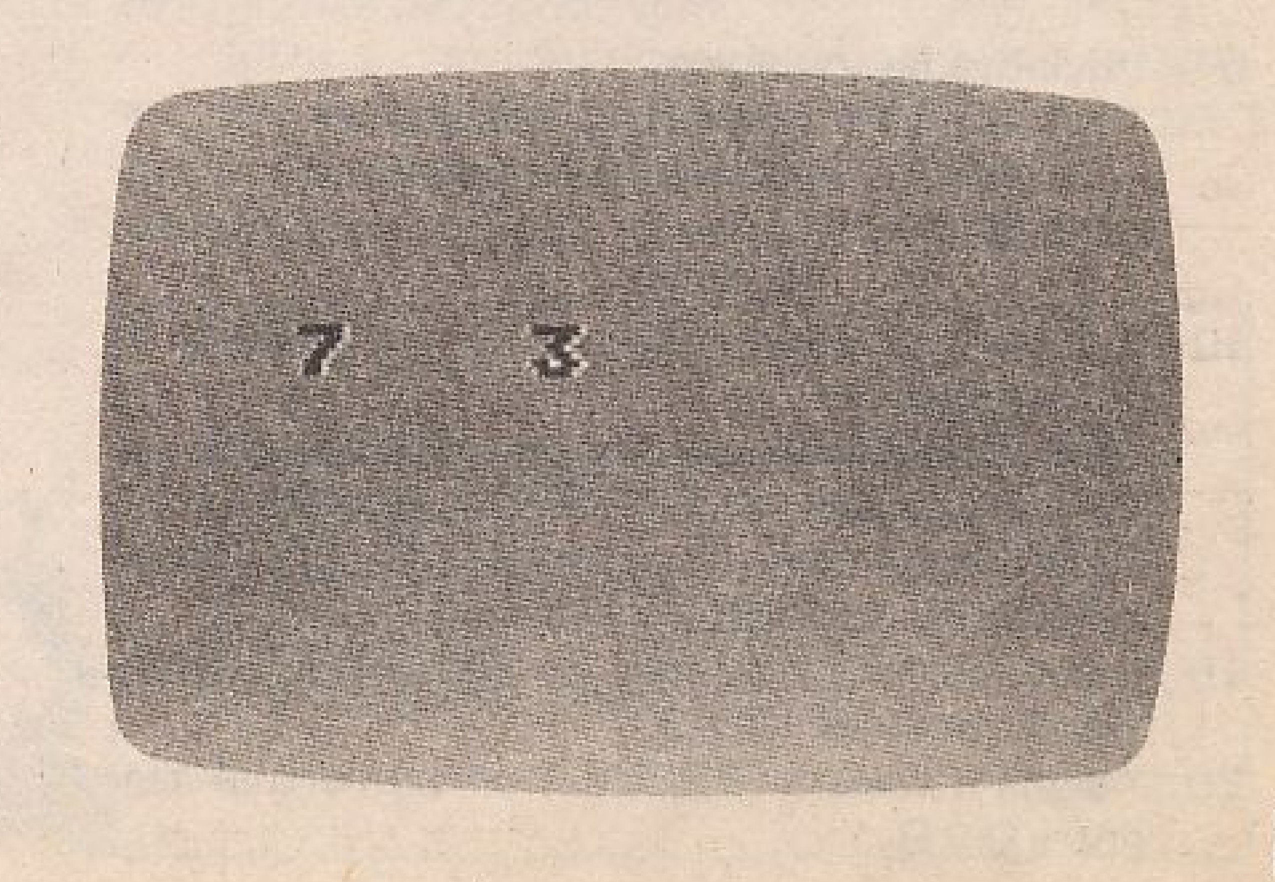
Esta sección es para los pequeños de la casa. Para poder interactuar con el ATARI necesitarán que uno de los mayores tipee, adapte o cree cortos programas que servirán para evaluar y mejorar habilidades.

RAPIDEZ DE REFLEJOS

Este programa permite medir la rapidez de los pequeños en asociar dos números iguales.

INDICACIONES DE USO

La pantalla, como indica la foto no entrega instrucciones. Es necesario practicar con el niño



para que entienda la interacción con el computador.

El pequeño debe presionar cualquier tecla sólo cuando dos números son iguales.

Si son diferentes debe esperar hasta que un nuevo par de números se le entregue en la pantalla. El cambio de color del fondo indica el momento.

Las respuestas incorrectas se descuentan de las correctas.

Al finalizar se proporciona una evaluación del uno al siete de acuerdo a sus aciertos.

Digite cuidadosamente el listado que se incluye a continuación:

```
5 PUNTO=0:CUENTA=1:MAXIMO=12
10 GRAPHICS 18:POKE 712,CUENTA*15:POKE
 708,0
15 POKE 20,0:POKE 19,0
20 PRIMER=INT(RND(0)*10)
30 SEGUNDO=INT(RND(0)*10)
40 POSITION 4,4:? #6;PRIMER
50 POSITION 8,4:? #6;SEGUNDO
60 POKE 764,255
70 IF PEEK(19)>1 THEN T=1:GOTO 100
80 IF PEEK(764)=255 THEN T=0:GOTO 70
85 IF PRIMER=SEGUNDO THEN PUNTO=PUNTO+
1:GOTO 150
90 PUNTO=PUNTO-1
100 IF PRIMER=SEGUNDO THEN PUNTO=PUNTO
-1:GOTO 150
110 PUNTO=PUNTO+1
150 REM
160 CUENTA=CUENTA+1
170 IF CUENTA>MAXIMO THEN 200
180 GOTO 10
200 GRAPHICS 18
210 POSITION 0,0:? #6;" CaLiFiCaCiOn
";
220 POSITION 3,3:? #6;PUNTO/MAXIMO*7
230 GOTO 230
```

Descripción del programa:

5 Inicializa las variables:
- PUNTO : Puntaje del participante
- CUENTA : Número de intentos
- MAXIMO : Número máximo de intentos (12)

10 Inicializa el modo gráfico 18, sin ventana de comunicación.
Modifica el color del fondo de acuerdo al valor de variable CUENTA (localización 712)
Modifica el color de números al negro (0) en la localización 708.

15 Inicializa localizaciones de control de tiempo (localizaciones 19 y 20).

20 Asigna a variable PRIMER un número al azar entre 0 y 9

30 Asigna a variable SEGUNDO un número al azar entre 0 y 9

40-50 Edita los números en pantalla

60 Inicializa la localización 764 para preparar el teclado en el momento que una tecla sea presionada

70 Establece si el tiempo para interactuar del usuario finalizó

80 Establece si el pequeño presionó alguna tecla

85 La condición que los números eran iguales se cumple e incrementa la variable PUNTOS por el acierto

90 Erró el usuario y por lo tanto se le descuenta

100 Erró también, pues no presionó las teclas y los números eran iguales

110 Acertó, pues los números eran diferentes

160 Incrementa la variable que cuenta el número de intentos

170 Condición de término con variable MAXIMO

180 Direcciona hacia un nuevo intento

200 Inicializa el modo gráfico 18: pantalla de calificación

210-230 Despliega nota del pequeño por su rendimiento.

POSIBLES MODIFICACIONES

1. Regular el tiempo para que se transforme en una medición de velocidad de reflejos. Para ello modifique línea 70:
   - Si desea más tiempo debe aumentar el valor (1)
   - Si desea menos tiempo debe recurrir a la localización 20. Puede partir de 240. Ej:

     IF PEEK(20) > 240 THEN T = 1: GOTO 100

     Si desea disminuir aún más, disminuya el valor 240 a 200, etc.
2. Aumentar un tercer número.
   Esto requiere crear otras líneas y modificar el programa.
   Es necesario precisar si el efecto de aprendizaje es el esperado en el pequeño.
3. Agregar una pequeña melodía de un cuento y tocarla cuando el pequeño acierta.

Esta sección tiene por objetivo demostrar que el ATARI es una herramienta eficaz para evaluar habilidades específicas a todo nivel.

El ingenio y la creación de los adultos pueden ayudar y estimular que los procesos de aprendizaje de los menores puedan proyectarse dentro de las características individuales del suyo.

Otra demostración es que no se necesita de grandes conocimientos de computación o de disponibilidad de tiempo para desarrollar pequeños programas que serán útiles.

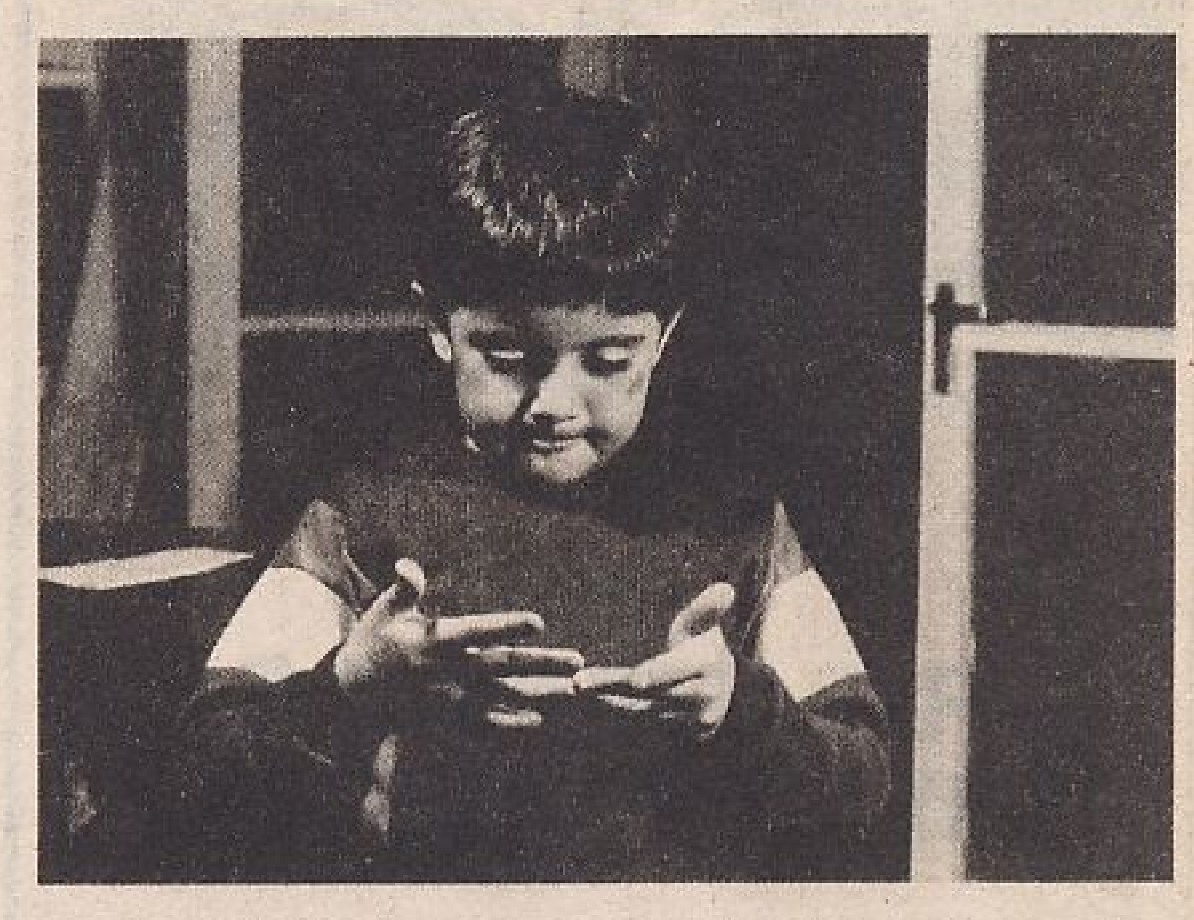
Los programas preparados para un caso particular tendrán en esta columna la posibilidad de extenderlo a todos los lectores de MUNDO ATARI

Clases con ATARI

Nuestra columna se dirige esta vez a los más pequeños de la casa, que son parte de la generación que utilizará extensamente la computación en el futuro.

La motivación para el tema de hoy fue dada por un pequeño, que al ser consultado: — ¿Cuánto es 8 más 5? comenzó con los dedos a contar:

```
10 REM PROGRAMA PARA SUMAR
15 CUENTA=0:ERROR=0
20 DIM ANS$(2)
30 PRINT CHR$(125)
40 K=INT(RND(0)*8)+1
45 LOC=40080
50 T=K:GOSUB 1000
60 POSITION 34,2:? K
80 POSITION 30,6:? "+"
100 M=INT(RND(0)*8)+1
110 LOC=40400
120 T=M:GOSUB 1000
130 POSITION 34,10:? M
150 POSITION 0,11:? "
" ;
200 RES=K+M
210 POSITION 2,14:? "  CUANTO ES ";K;"
 + ";M
215 POSITION 0,15:? "              ";
220 POSITION 0,15:INPUT ANS$
230 TRAP 210
240 USTED=VAL(ANS$)
250 IF USTED=RES THEN GOTO 1200
260 ERROR=ERROR+1
270 GOTO 210
1000 FOR J=1 TO T
1010 INC=J*2
1020 POKE LOC+INC,84
1030 NEXT J
1040 RETURN
1200 CUENTA=CUENTA+1:? CHR$(253)
1210 IF CUENTA=21 THEN GOTO 2000
1220 GOTO 30
2000 ? CHR$(125):? CHR$(253)
2030 POSITION 10,2:? "RESULTADOS"
2040 POSITION 2,10:? "ACIERTOS    = ";CU
ENTA
2050 POSITION 2,15:? "ERRORES     = ";ER
ROR
2060 END
2070 GOTO 30
```



— Uno, dos, tres, cuatro, cinco, seis, siete, ocho... luego quiso adicionar los otros cinco, pero... se encontró con que le faltaban dedos para terminar.

Pensando en el problema, encontramos que este pequeño tuvo problemas para identificar y asociar el número con la palabra.

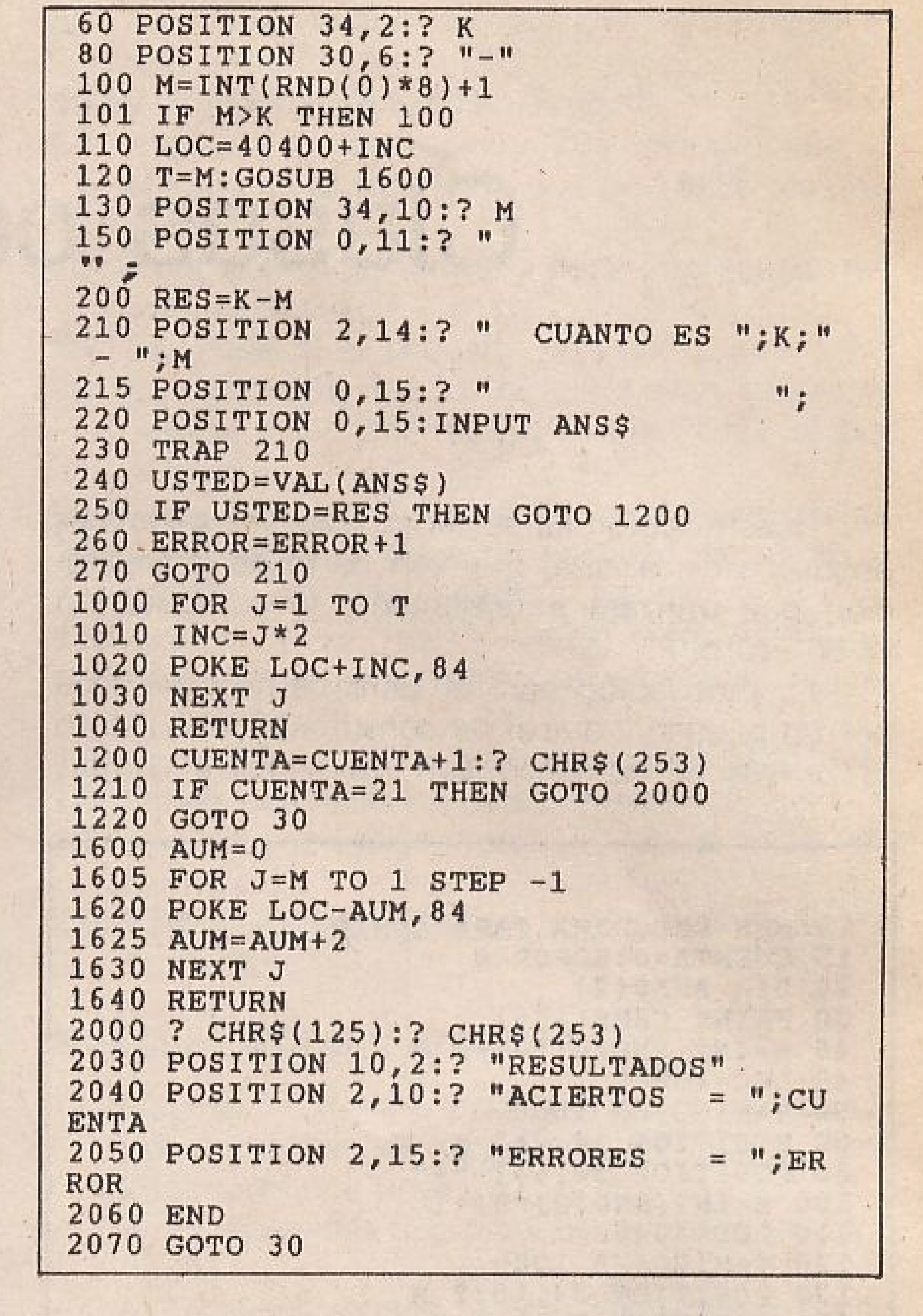
Pregunté a mi ATARI: ¿Eres capaz de ayudar a mi amiguito?. El está siempre dispuesto, de modo que con unos minutos de gasto de electricidad aparecen dos programas simples creados para este menester.

El listado 1 suma números que representa además mediante figuras.

El listado 2 resta números en que el efecto visual importante es la ubicación de la representación gráfica de los números.

Liste a continuación los programas y pruébelos con los peques de la casa. Modifíquelos para que puedan servir en otras aplicaciones.

```
10 REM PROGRAMA PARA RESTAR
15 CUENTA=0:ERROR=0
20 DIM ANS$(2)
30 PRINT CHR$(125)
40 K=INT(RND(0)*8)+1
45 LOC=40080
50 T=K:GOSUB 1000
```



```
60 POSITION 34,2:? K
80 POSITION 30,6:? "-"
100 M=INT(RND(0)*8)+1
101 IF M>K THEN 100
110 LOC=40400+INC
120 T=M:GOSUB 1600
130 POSITION 34,10:? M
150 POSITION 0,11:? "
" ;
200 RES=K-M
210 POSITION 2,14:? "  CUANTO ES ";K;"
 - ";M
215 POSITION 0,15:? "                    ";
220 POSITION 0,15:INPUT ANS$
230 TRAP 210
240 USTED=VAL(ANS$)
250 IF USTED=RES THEN GOTO 1200
260 ERROR=ERROR+1
270 GOTO 210
1000 FOR J=1 TO T
1010 INC=J*2
1020 POKE LOC+INC,84
1030 NEXT J
1040 RETURN
1200 CUENTA=CUENTA+1:? CHR$(253)
1210 IF CUENTA=21 THEN GOTO 2000
1220 GOTO 30
1600 AUM=0
1605 FOR J=M TO 1 STEP -1
1620 POKE LOC-AUM,84
1625 AUM=AUM+2
1630 NEXT J
1640 RETURN
2000 ? CHR$(125):? CHR$(253)
2030 POSITION 10,2:? "RESULTADOS"
2040 POSITION 2,10:? "ACIERTOS   = ";CU
ENTA
2050 POSITION 2,15:? "ERRORES    = ";ER
ROR
2060 END
2070 GOTO 30
```

Comentarios al listado:

Variable LOC: se relaciona con las localizaciones de memoria que corresponden a la pantalla en modo gráfico 0.

El puntero para la primera posición de pantalla se encuentra registrado en las localizaciones 88 y 89.

ACTIVIDAD:

Encienda su ATARI con BASIC incluído y digite en modo directo:

PRINT PEEK(88) +256 * PEEK(89)

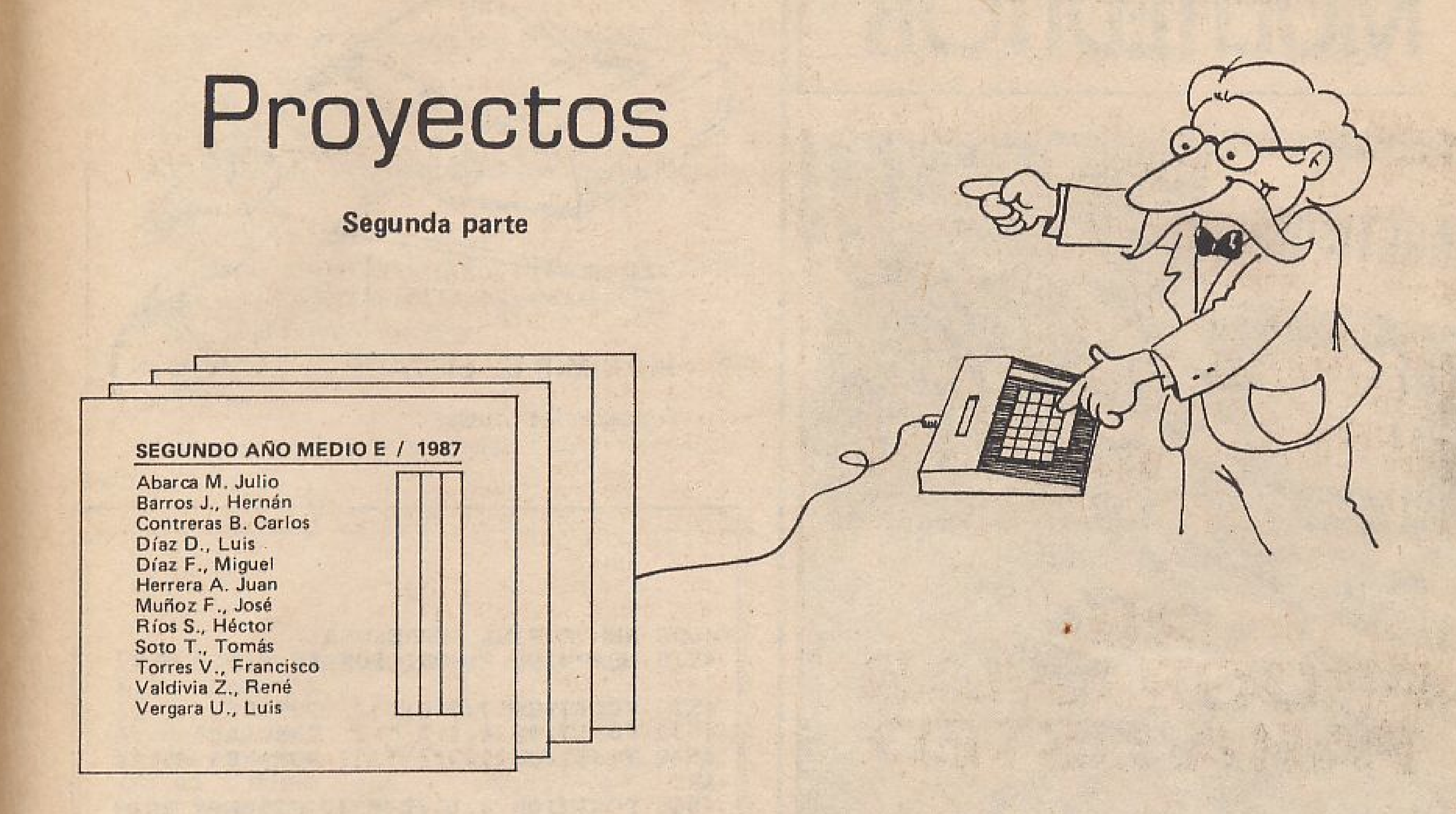
En el caso del ATARI 800 XL ó 130 XE el valor de retorno es 40000.

Con esta información puede usted adaptar el programa para el ATARI 600 XL. Al valor retornado por el 600 XL adicione las cantidades equivalentes: 80, 400.

Para más información al respecto ver en la columna DE BYTE EN BYTE de este mes las localizaciones 87, 88 y 89.

Proyectos

Segunda parte



La columna PROYECTO continúa este mes con la etapa de definición de estructuras para el ingreso de datos.

El primer problema a solucionar es el **listado de alumnos del curso.** Este listado debe contemplar como mínimo las siguientes opciones:

A. Inicializar el listado en desorden.
B. Ordenar el listado (hacer lista del curso).
C. Buscar un nombre en la lista.
D. Imprimir el listado en la impresora.
E. Modificar un nombre del listado.
F. Ingresar un nuevo nombre a la lista base.
G. Grabar listado de nombres.
H. Cargar el listado de nombres.

En el funcionamiento de un colegio estas listas son una necesidad fundamental para la organización del curso.

El tener acceso rápido a dichas listas es de vital importancia.

Ahora, con la idea de que el funcionamiento del colegio implica un **sistema,** esta lista se almacenará en forma de archivo independiente en el periférico de memoria permanente (casetera, diskettera).

La columna TORPEDO nos proporciona material inicial para abordar nuestro problema. Observe la estrategia y el método de resolución de los problemas. El reunir material en forma de pequeños bloques, rutinas o subrutinas hacen más fácil y comprensible el desarrollo de una programación.

Tomemos como base este programa y modifique en línea 20:

N = 50 Número máximo de alumnos.
E = 30 Se mantiene caracteres máximos para el nombre de un alumno.

La línea 30 modifica a DIM:

```
30 ARRE$(A + 2), ELEM$(E), NUM$(2), RES$(1)
```

Agregamos dos bytes al arreglo para guardar el número de alumos del curso.

```
4020 TRAP 4020: PRINT "Digite número de alumnos de su curso": INPUT NUM$: N = VAL(NUM$)
4025 IF N >50 THEN GOTO 4020
4095 ARRE$(A + 1, A + 2) = NUM$
```

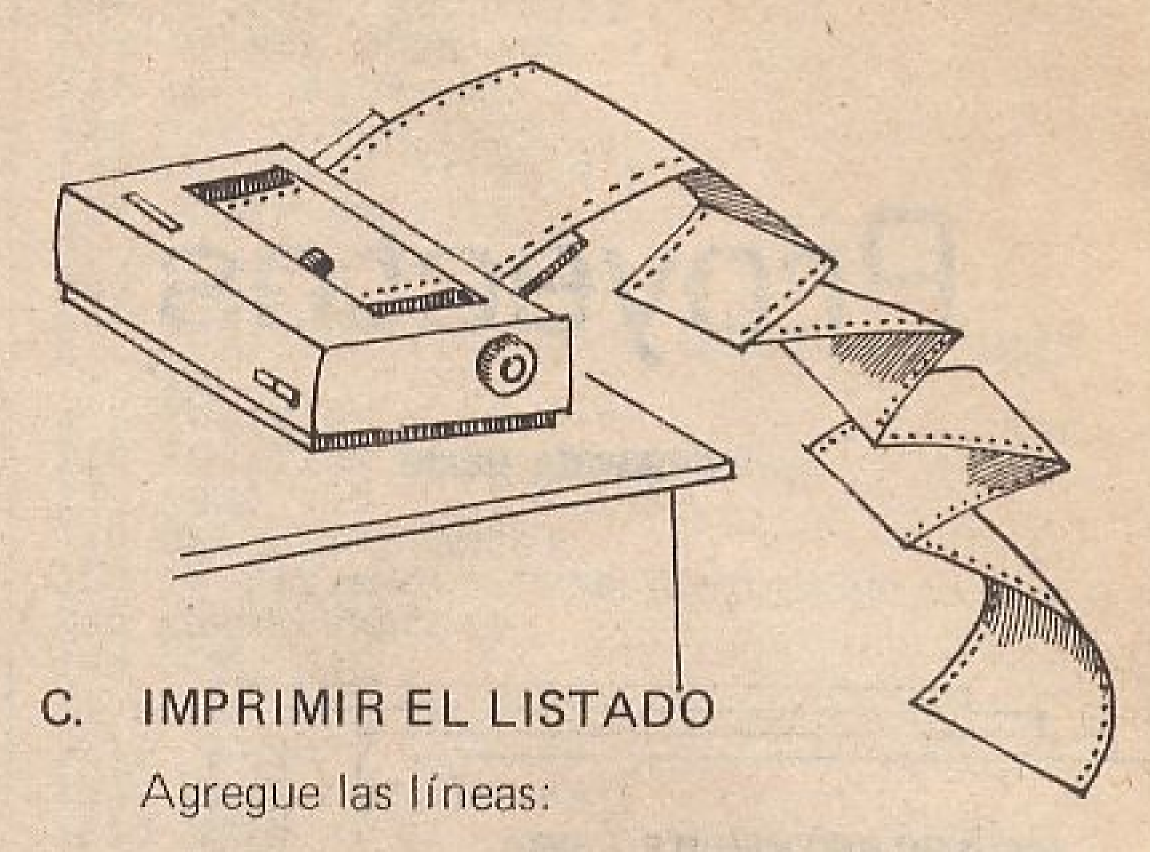
Esta rutina nos cubre dos opciones:

1. Inicializar el listado en desorden.
2. Buscar un elemento de la lista.

¿Cómo solucionamos el problema para las otras opciones?

B. ORDENAR EL LISTADO

Se requiere de una rutina que ordene los elementos del string en forma alfabética . La rapidez puede estar dada por una rutina binaria.



C. IMPRIMIR EL LISTADO

Agregue las líneas:

```
4500 REM RUTINA IMPRESORA
4510 GRAPHICS 0:POSITION 10,2:? " MENU
"
4520 POSITION 4,5:? " 1  TOTAL"
4530 POSITION 4,7:? " 2  PARCIAL"
4540 POSITION 4,9:? " 3  NOMBRES AL AZ
AR"
4550 POSITION 4,11:? " 4  RETORNA AL M
ENU"
4600 POSITION 0,16:? " DIGITE el numer
o que selecciona "
4610 INPUT M:IF INT(M)<1 OR M>4 THEN 4
600
4620 REM DEBE IR A CARGAR LISTADO DEL
PERISFERICO QUE USE
4650 ON M GOTO 4700,4800,5000
4700 NUM=VAL(ARRE$(A+1,A+2))
4710 FOR J=1 TO NUM
4720 K=J:GOSUB 4300
4730 LPRINT ELEM$
4740 NEXT J
4750 GOTO 4500
4800 GRAPHICS 0
4810 PRINT "Digite numero de clave par
a primer        alumno lista"
4820 INPUT PRIMER
4830 PRINT "Digite numero de clave par
a ULTIMO        alumno lista"
4840 INPUT ULTIMO
4850 FOR J=PRIMER TO ULTIMO
4860 K=J:GOSUB 4300
4870 LPRINT ELEM$
4880 NEXT J
4890 GOTO 4500
4900 GRAPHICS 0
4910 PRINT "Digite numero de alumnos q
ue desea        imprimir    "
4915 INPUT NUM
4920 FOR J=1 TO NUM
4930 PRINT "Digite numero de clave par
a alumno"
4940 INPUT K
4950 GOSUB 4300
4960 LPRINT ELEM$
4970 NEXT J
4980 GOTO 4500
```

D. MODIFICAR UN NOMBRE DE LISTA

Para modificar el nombre, las etapas a seguir son las siguientes:

1. Cargar lista de alumnos del periférico de memoria permanente.
2. Ingresar el número de clave para alumno que modifica.
3. Validar el número dentro de los valores permitidos.
4. Buscar el nombre.
5. Confirmar si el nombre es el que se modifica.
6. Retorno si existe error de ingreso de clave.
7. Ingresar el nuevo nombre.
8. Ingresarlo al arreglo de string.
9. Retornar.

El listado siguiente contiene las etapas formuladas:

```
5100 REM RUTINA MODIFICA UN NOMBRE.
5105 GRAPHICS 0
5110 REM carga lista del medio fisico
de grabacion
5120 POSITION 2,2:? "DIGITE la clave p
ara el NOMBRE  que       modifica"
5130 TRAP 5120:INPUT K
5140 N=VAL(ARRE$(A+1,A+2))
5150 IF INT(K)<1 OR K>N THEN ? "ERROR
DE RANGO":GOTO 5120
5160 GOSUB 4300
5170 ? :? ELEM$
5180 ? "Corresponde al elemento a modi
ficar      DIGITE S/N "
5190 INPUT RES$
5200 IF RES$="N" OR RES$="n" THEN GRAP
HICS 0:GOTO 5120
5210 ? :? "DIGITE el nombre modificado
"
5220 INPUT ELEM$
5230 L=LEN(ELEM$)
5240 INC=(K-1)*E
5250 ARRE$(1+INC,E+INC)=ELEM$
5260 RETURN
```

La opción **Ingresar un nuevo nombre a la lista** será asignada a usted para desarrollarla como tarea para el próximo número. Ocupe las líneas 5300 a 5500. Elija como modelo la opción Modifica.

Indique las etapas en palabras y a continuación programe las líneas correspondientes.

Para la opción **B. Ordenar el listado,** nos ayudará la columna TORPEDO del próximo número: nos proporcionará una rutina simple para ordenar elementos de un string.

Para las opciones G y H. **Grabar y Cargar archivo de listado de nombres** nos remitiremos a las columnas especializadas: DOMINANDO LA 1050 y CONTROLANDO EL CASSETTE.

Primero grabe el programa residente en memoria con un nombre en su periférico disponible (casetera o diskettera). Utilice para ello:

SAVE "D:PROYECTO.MAY" (diskettera)
CSAVE (casetera)

Ahora pruebe las posibilidades que tiene con este fragmento de programa:

1. Ejecute con RUN y podrá crear un listado de alumnos para su curso y buscar un determinado nombre por la clave.
2. Digite GOTO 4500 y podrá seleccionar una impresión si posee una impresora.
3. Digite 5100 y podrá modificar uno de los nombres de su lista original.

Qué poco práctico resulta tener que recordarse la línea, detener el programa y ejecutar un GOTO cada vez que se desea ejecutar una rutina.

Para optimizar esta situación existen los MENU de selección y rutinas para seleccionar un ítem del menú en forma rápida.

Esta será también una tarea para usted: proporcionar una rutina para seleccionar el menú.

El próximo mes completaremos el programa **listado de alumnos del curso.**

Profesor ATARIN

ERRATA

Queremos advertir a nuestros lectores de un error que se deslizó en el número 0 de MUNDOATARI.
El programa No. 1 de la columna Aplicaciones: Clases con ATARI, página 36, se desarrolla por medio de un ciclo de 1 a 10, el que quedó abierto por omisión de la instrucción NEXT J.
Para solucionar dicho error rogamos digitar la siguiente línea:

65 NEXT J

Agradecemos especialmente la gentil colaboración de dos lectores: Srta. Gabriela Tapia A. y Sr. Héctor Alegría A. ambos de la ciudad de Concepción, quienes nos advirtieron acertadamente al respecto. A ellos y a nuestros muchos lectores que nos han escrito les hemos respondido en forma privada. Gracias por su estímulo.

OBSERVACION: **Nos gustaría que al enviar cartas a la redacción nos proporcionaran más información acerca de ustedes: edad, ocupación, configuración de equipo, necesidades de software, etc.**



Esta sección atenderá la creciente demanda de información por la línea ST de ATARI

ATARI 1040 ST

Hoy en día el mercado de los computadores personales está regido principalmente por una enorme competencia de cualidades versus precio, y como ya es costumbre, ATARI tiene mucho que decir.

Hace aproximadamente un año y medio atrás ATARI hizo noticia con el lanzamiento de su computador estrella, el ATARI 520 ST, asombrando al mundo con un equipo de características espectaculares y con un precio muy por debajo de lo usual. Luego de unos pocos meses logró hacernos ver que no habíamos perdido nuestra capacidad de asombro al lanzar al mercado el hermano mayor del 520 ST, el ATARI 1040 ST, el primer computador del mundo de 1 Megabyte de RAM, con un precio accesible. Esto le ha valido el título en Europa del Computador del Año 1986.

Este título no es injustificado, pensemos que estamos hablando de un equipo que tiene la capacidad de memoria de un

Main Frame de aproximadamente US $ 30.000, posee un procesador tanto o más veloz que los de computadores que hasta hace pocos años se usaban para manejar la información de un banco. A esto agreguémosle una tremenda facilidad de uso y una completa gama de programas para las más diversas aplicaciones. Es indudable que estamos hablando de un equipo capaz de hacer cosas que son difíciles de imaginar usando los métodos tradicionales, pero que aún así se nos presenta a un precio realmente razonable si lo comparamos con equipos de similares características.

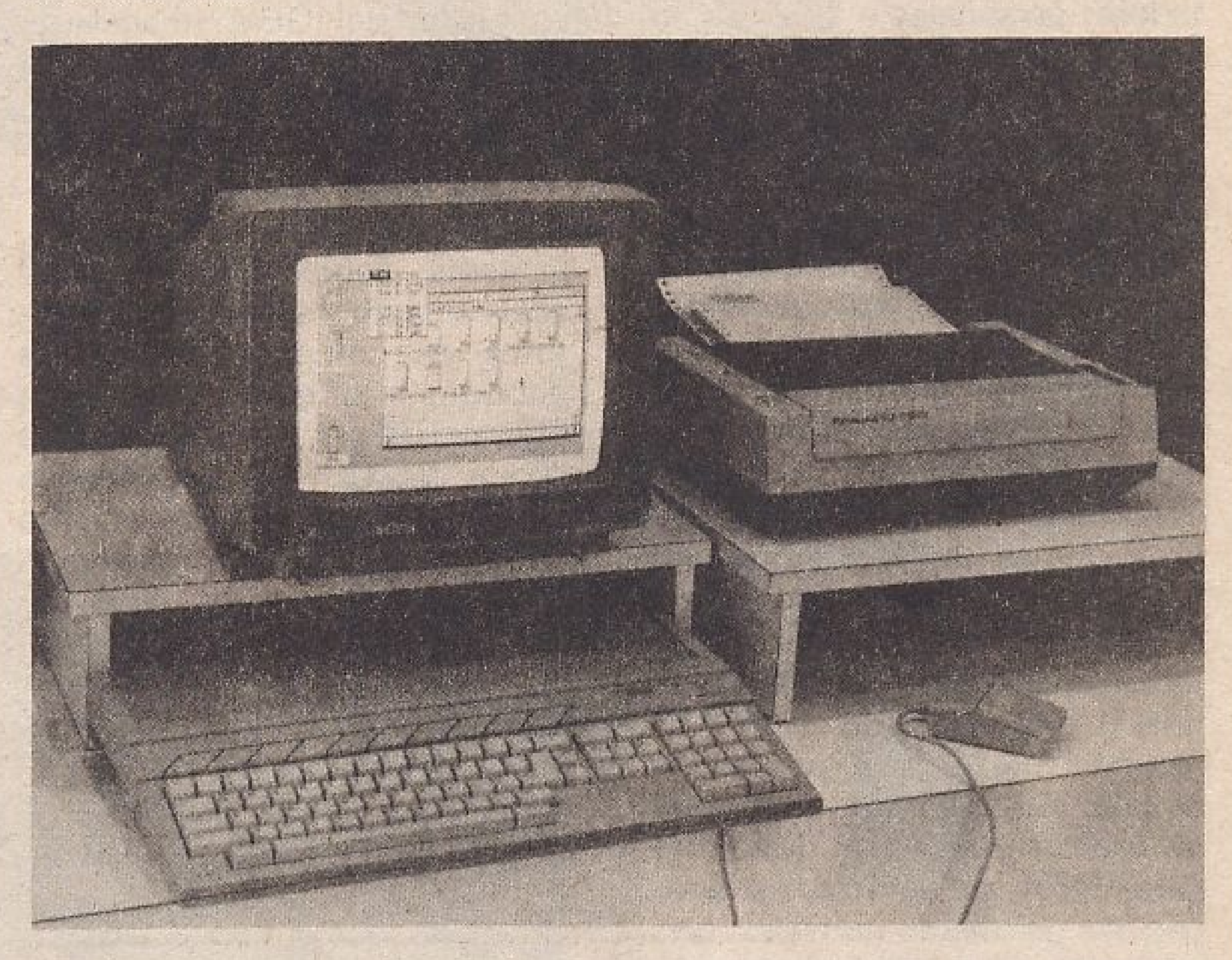
Computador para todos

Al igual que en Europa, el entusiasmo demostrado por el mercado norteamericano ha sido notable, alcanzando en el primer año de existencia del ATARI 1040 ST la cifra record de 200.000 unidades vendidas. La revista **Byte**, vocero de la industria computacional, calificó este equipo como ***el computador para todos***, agregando que ***el 1040 ST es una de las piedras angulares de la computación personal.*** Esta última frase se refiere a que con el ATARI 1040 ST se ha puesto al alcance de la mayoría de las personas un computador de características que antes estaban reservadas a equipos de alto costo y por consiguiente a muy pocas personas.

En cuanto a las características técnicas, éstas son prácticamente iguales a las del computador ATARI 520 ST, salvo tres diferencias de consideración. La primera es el volumen de memoria RAM que asciende hasta 1024 Kbytes, por lo tanto el ATARI 1040 ST cuenta con exactamente el doble de la memoria de su hermano menor. Además este computador cuenta con una diskettera de doble lado y 720 Kbytes de almacenamiento incorporada en la consola, lo que evita los cables externos. Esta diskettera es equivalente a la diskettera ST 314. Por último, la tercera diferencia esta dada por la ausencia del modulador de radio frecuencia (conexión a televisor). Esto se debe a que en ATARI Corp. determinaron que dadas las capacidades superiores de este equipo, las mayores ventas se darían dentro de las aplicaciones profesionales, como estaciones de trabajo, donde el uso prolongado del equipo requiere de una pantalla de alta calidad que disminuye el cansancio de la vista.

Las diferencias anteriores no afectan el sistema operativo, por lo tanto, ambos equipos son absolutamente compatibles en lo que respecta al software, sólo que el ATARI 1040 ST permite un mayor manejo de datos en memoria.

El ATARI 1040 ST no sólo resuelve los mismos problemas que los otros computadores resuelven, además resuelve el problema que los otros computadores crearon: **EL PRECIO.**



CARACTERISTICAS TECNICAS

ARQUITECTURA
- RAM standard de 1024 Kb
- CPU Motorola 68000
- CPU de 16/32 bits
- Reloj de 8 MHz
- ROM de 192 Kbytes
- Sistema Operativo TOS combinado con CP/M 68 K y GEM
- Mouse de dos botones
- Cartridge de hasta 192 K

MEDIOS DE ALMACENAMIENTO
- Unidad de diskette está incorporada
- Diskettes de 3 1/2"
- Capacidad 360 Kb ó 720 Kb
- Port Disco Duro
- Capacidad 20, 30 ó 60 Mb
- Velocidad Disco Duro: 10 Mb/seg.

GRAFICOS
- Alta resolución monocromo 640 X 200
- Media resolución color 640 X 200
- Baja resolución color 320 X 200
- Paleta de 512 colores

TECLADO
- Ergonómico tipo QWERTY
- 94 teclas (10 de función)
- Teclado numérico separado
- Teclas de cursor
- Procesador independiente
- Regulación de presión
- Regulación de respuesta

SONIDO
- Tres canales
- Cuatro octavas
- Control de sintetizadores

PUERTAS DE ENTRADA Y SALIDA
- Impresora CENTRONICS
- Port RS232-C
- Floppy Disk 5 1/4"
- Floppy Disk 3 1/2"
- DMA (Hard Disk)
- MIDI (Instrumentos musicales)
- Mouse
- Joystick
- Cartridge

Planilla electrónica VIP GEM

Cuando hablamos de computadores personales o PCs., por lo general se los asocia con un programa de gran divulgación, el LOTUS 123, programa que ha motivado por lo menos la compra del 10% de los computadores IBM o IBM compatibles en el mundo entero, y esto con justa razón, ya que se trata de un poderosísimo programa que realiza tres funciones básicas muy usadas en computación, la primera es la función de planilla electrónica, que es una hoja de cálculo formada por **celdas** que puede ser ocupada por números u operaciones matemáticas en base a una o más de estas celdas. La segunda función que realiza es la de base de datos, o la capacidad de ordenar datos por diferentes parámetros, que asociada a la capacidad de cálculo numérico permite crear una infinidad de aplicaciones como: inventarios, flujos de caja, sistemas de cuentas corrientes, o tal vez generar un menú electrónico que le permita saber cuantas calorías tendría un plato variando sus ingredientes. Y por último cuenta con la facilidad de generar gráficos para poder visualizar más fácilmente los datos con que se está trabajando.

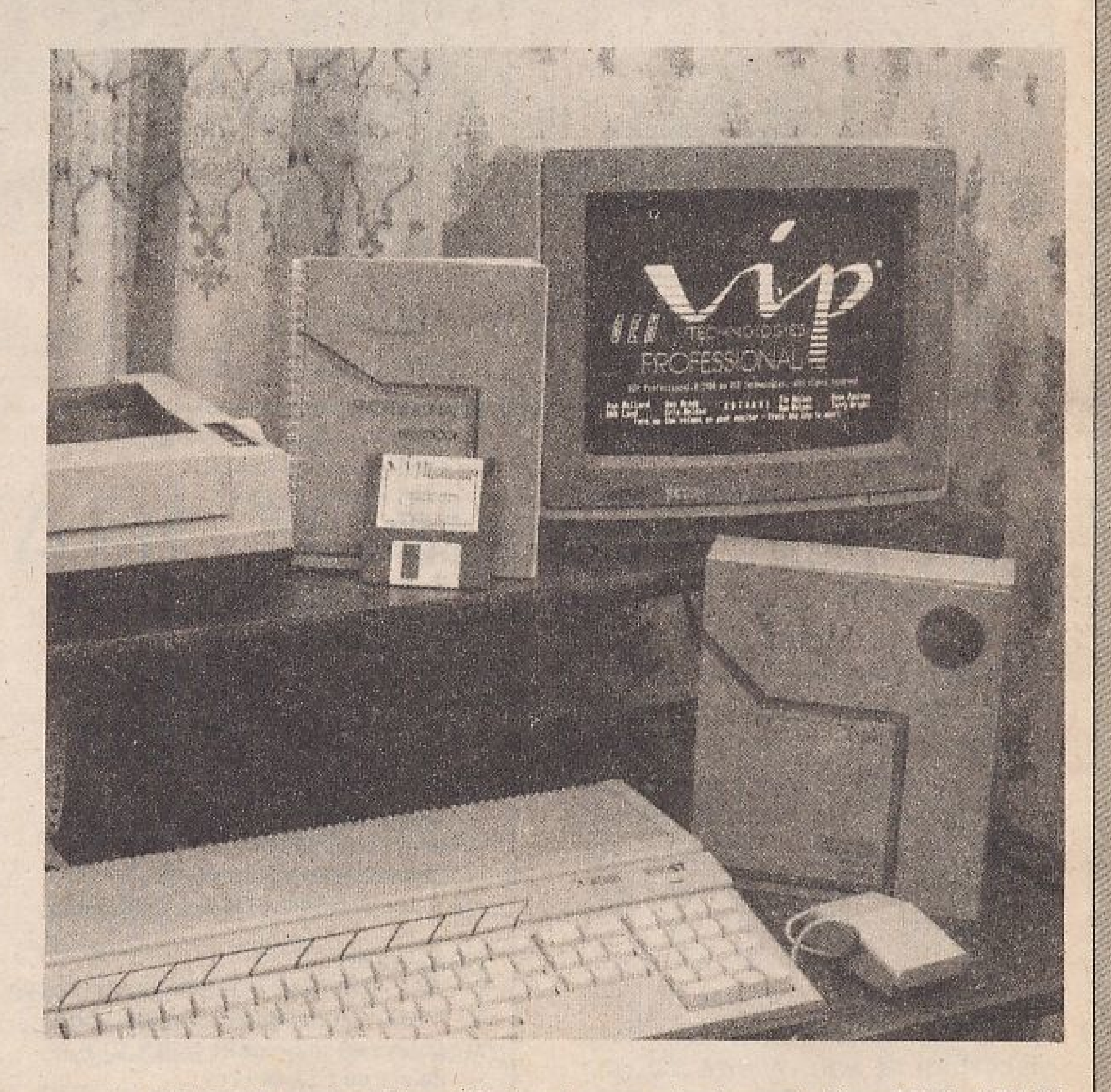
¿LOTUS o VIP?

Tal vez usted se preguntará por qué hablamos tanto del LOTUS 123 si este es un artículo sobre el VIP PROFESSIONAL GEM. Esto se debe a que ambos programas hacen lo mismo, y no sólo eso, pero con ligeras diferencias. La primera es la capacidad, el VIP posee cuatro veces más capacidad que el LOTUS 123, y además usa todas las capacidades del GEM (manejador de ambiente gráfico) cosa que simplifica tremendamente el uso del programa.

Una de las cosas que caracteriza al LOTUS es su uso medianamente complicado, si usted observa la publicidad de los institutos de computación, en las publicaciones especializadas en PCs. podrá ver gran cantidad de cursos de LOTUS que pueden durar desde pocas semanas hasta varios meses. Con la implementación del GEM en el VIP usted podrá hacer en dos días lo mismo que haría en veinte si usara el LOTUS. Esto se debe a que en el GEM el uso del programa es intuitivo, o sea, el usuario busca la instrucción que desea ejecutar en lugar de memorizarla, utilizando así la memoria visual y mecánica además de la memoria abstracta.

Además, y como si esto fuera poco, una ventaja adicional de VIP PROFESSIONAL GEM, es la capacidad de trabajar como procesador de texto, lo que permitiría por ejemplo emitir cartas seleccionando entre sus clientes a los deudores y emitirles cartas con el cálculo de los intereses desde su último vencimiento (todo esto en forma automática).

Sin duda el VIP es una de las más poderosas y versátiles herramientas disponibles hoy día para los computadores ATARI 1040 ST y 520 ST. ●



VIP GEM: Aplicación profesional a otro nivel para la línea ST de ATARI



PANASONIC Y ATARI SE ENTRETIENEN JUNTOS...

...Y cómo lo hacen!

Desde el momento en que la Impresora KX-P1081 de PANASONIC se dio cuenta que era absolutamente compatible con el microcomputador ATARI, se lo pasan juntos todo el día.

La Impresora KX-P1081 de PANASONIC ya tiene experiencias anteriores, y siempre demostró ser realmente compatible.

Ahora, con el microcomputador ATARI está nuevamente feliz: grafica todo el día, comprime y hace las veces de una perfecta secretaria con una tipografía que es todo un lujo!

Y si quiere saber aún más, imprime listados a 120 CPS en su modo DRAFT, obteniendo su correspondencia en modo NEAR LETTER QUALITY a 24 CPS, y cuadros en modo COMPRESSED a 102 CPS, todo con una absoluta nitidez. No en vano la llaman la rápida y perfecta!

Pregunte por cualquiera de las Impresoras PANASONIC que hay en nuestro mercado (la KX-P1082, KX-P1592 ó la KX-P1595), y verá que todas son 100% compatibles con los microcomputadores ATARI.

Esta confiabilidad y durabilidad, y una calidad a toda prueba, asegurán una relación perfecta... y por mucho tiempo.

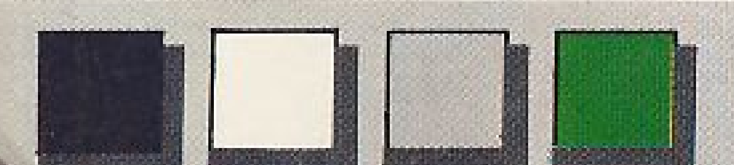
Línea de Impresoras PANASONIC para satisfacer todas las necesidades, incluso la más exigentes.

Un acierto tecnológico que sintetizá todas las características que en otros son opcionales.

AUTOMATIZACION DE OFICINAS

MELLAFE Y SALAS M.R.

ANTOFAGASTA: *COMERCIAL PRAT, Latorre 2500 - F. 223083.* **SANTIAGO:** *COELSA COMPUTACION, Vicuna Mackenna 1705 - F. 5566006. MULTIMATICA, San Antonio 73 - F. 382663. C.J. COMUNICACIONES, Av. L.B. O'Higgins 1146 - F. 727355. MAICOM. Eliodoro Yañez 2675 - F. 2233338. COMPUTER MARKET, Pueblo del Ingles Local 66 - F. 2243474.* **CONCEPCIÓN:** *CRECIC S.A., Los Acacios 107 - F. 371317-371417. Galeria Internacional. L. 24/25 - F. 225754.* **TEMUCO:** *COMPUMATICS, M. Montt 730 - F. 234239.* **OSORNO:** *COMPUMATICS, Ramirez 870 - F. 5757.*



LOS COLORES INTELIGENTES.

AUTOMATIZACION DE OFICINAS, LA SUMA DE TODAS LAS VENTAJAS.

Dele a su Computador Atari el lugar que se merece.

1. Mueble Comp. Americano (7519/10).



2. Mesa TV Video Comp, con bandeja deslizable (7523/11).

3. Secreter, con puerta abatible y llave para proteger los componentes de su computador (7528/11).



4. Escritorio Melody, con bandeja deslizable (7580/11).

5. Centro modular, compuesto de escritorio base (7527/11), porta TV (7526/11), repisa lateral (7525/11), cajón (7521/10) y mesa impresora (7524/11).



Productos con el sello de calidad CIC.



CICFORMATICA

LINEA DE MUEBLES PARA COMPUTADORES PERSONALES.



Adquiera estos modelos en las Tiendas CIC-Hogar, o en los principales Distribuidores del país.